Analysis of Women's E-Commerce Clothing Reviews

Sabrina Lee | Yun Xiao | Maggie Fan | Daisy Wang | Sammi Chueh

Medium:

https://www.linkedin.com/pulse/analysis-womens-e-commerce-clothing-reviews-yun-xiao/?trackingId=asFB%2BQJQTTOuPdaix23csQ%3D%3D

Code:

https://github.com/StellaYunX/women_clothing_e_commerce_reviews.git

1. Project Description and Objectives

For our project, we are going to use the dataset of women's e-commerce clothing reviews to perform analytics. Our research can be split into three parts. First, we are looking into the relationship between classes / departments and recommendations to see what kind of classes / departments got relatively positive or negative reviews. Second, we are going to conduct sentiment analysis and produce word clouds to look into positive and negative words. Lastly, we are going to build different predictive models and see how the models perform.

2. Data Descriptions

We are using the dataset: **Women's E-Commerce Clothing Reviews.**

https://www.kaggle.com/datasets/nicapotato/womens-ecommerce-clothing-reviews?select=Womens+Clothing+E-Commerce+Reviews.csv

This dataset is real commercial data, it includes 11 feature columns and 23486 rows. Every row represents a customer review.

	Unnamed: 0	Clothing ID	Age	Title	Review Text	Rating	Recommended IND	Positive Feedback Count	Division Name	Department Name	Class Name
0	0	767	33	NaN	Absolutely wonderful - silky and sexy and comf...	4	1	0	Initmates	Intimate	Intimates
1	1	1080	34	NaN	Love this dress! it's sooo pretty. i happene...	5	1	4	General	Dresses	Dresses
2	2	1077	60	Some major design flaws	I had such high hopes for this dress and reall...	3	0	0	General	Dresses	Dresses
3	3	1049	50	My favorite buy!	I love, love, love this jumpsuit. it's fun, fl...	5	1	0	General Petite	Bottoms	Pants
4	4	847	47	Flattering shirt	This shirt is very flattering to all due to th...	5	1	6	General	Tops	Blouses

Data Preprocessing

We then drop unnecessary columns and rename the rest of the columns for further analysis.

```
# Drop Unnecessary Columns
data.drop(labels =['Unnamed: 0'],axis = 1,inplace = True)

# Rename Columns
data = data.rename(columns={'Clothing ID': 'Clothing_ID','Review Text':
'Review_Text','Recommended IND': 'Recommended_IND',
'Positive Feedback Count': 'Positive_Feedback_Count', 'Division Name':
'Division_Name',  'Department Name': 'Department_Name', 'Class Name':
```

```
'Class_Name'})

# Show Dataframe
data.head()
```

	Clothing_ID	Age	Title	Review_Text	Rating	Recommended_IND	Positive_Feedback_Count	Division_Name	Department_Name	Class_Name
0	767	33	NaN	Absolutely wonderful - silky and sexy and comf...	4	1	0	Initmates	Intimate	Intimates
1	1080	34	NaN	Love this dress! it's sooo pretty. i happene...	5	1	4	General	Dresses	Dresses
2	1077	60	Some major design flaws	I had such high hopes for this dress and reall...	3	0	0	General	Dresses	Dresses
3	1049	50	My favorite buy!	I love, love, love this jumpsuit. it's fun, fl...	5	1	0	General Petite	Bottoms	Pants
4	847	47	Flattering shirt	This shirt is very flattering to all due to th...	5	1	6	General	Tops	Blouses

Text Processing

For text processing, first we remove all the punctuation and special characters in the reviews. By doing so, we can preclude those buzz words, which is a preparation for further classifying. Next, we convert all the uppercase words into lowercase words. Finally, we remove the stopwords, which are also considered buzz characters in this case.

```
#Regular Expression
import re
def sentence_rge(data):
    data = re.sub('<[^>]*>','',str(data))
    data = re.sub('[\W]+',' ',data.lower())
    return data

data_train['Review_Text'] = data_train['Review_Text'].apply(sentence_rge)


# Stopwords
data_train['Review_Text'] = data_train['Review_Text'].apply(lambda x: '
'.join([word for word in x.split() if word not in (stopwords)]))
data_train.head()
```

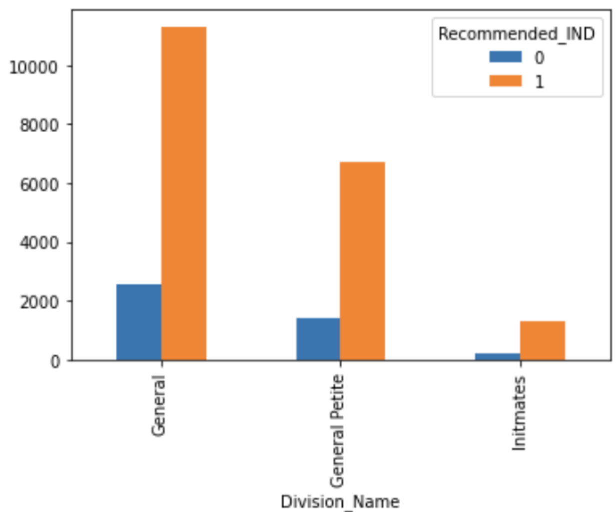
	Clothing_ID	Age	Title	Review_Text	Rating	Recommended_IND	Positive_Feedback_Count	Division_Name	Department_Name	Class_Name
0	767	33	NaN	absolutely wonderful silky sexy comfortable	4	1	0	Initmates	Intimate	Intimates
1	1080	34	NaN	love dress s sooo pretty happened find store m...	5	1	4	General	Dresses	Dresses
2	1077	60	Some major design flaws	high hopes dress really wanted work initially ...	3	0	0	General	Dresses	Dresses
3	1049	50	My favorite buy!	love love love jumpsuit s fun flirty fabulous ...	5	1	0	General Petite	Bottoms	Pants
4	847	47	Flattering shirt	shirt flattering due adjustable front tie perf...	5	1	6	General	Tops	Blouses

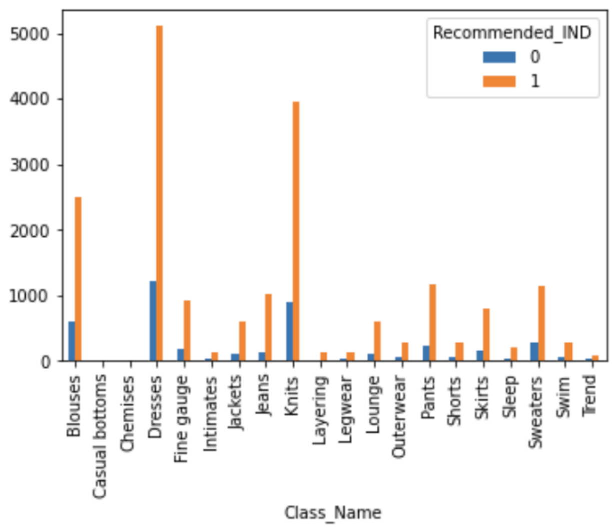
Data Exploration

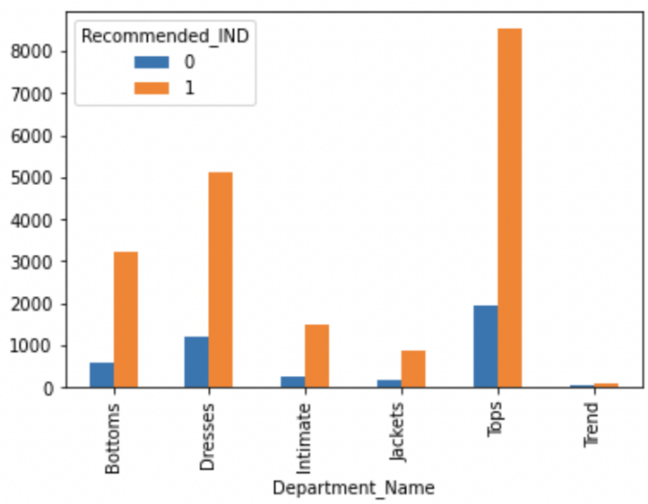
By performing exploratory data analysis, we examine the distribution of positive and negative reviews by different types of clothes, division, ratings, and so on. The way we tell if this review is positive or not is based on whether a customer would recommend the product. “Recommended_IND” is a binary variable stating where the customer recommends the product where 1 is recommended, while 0 is not recommended.

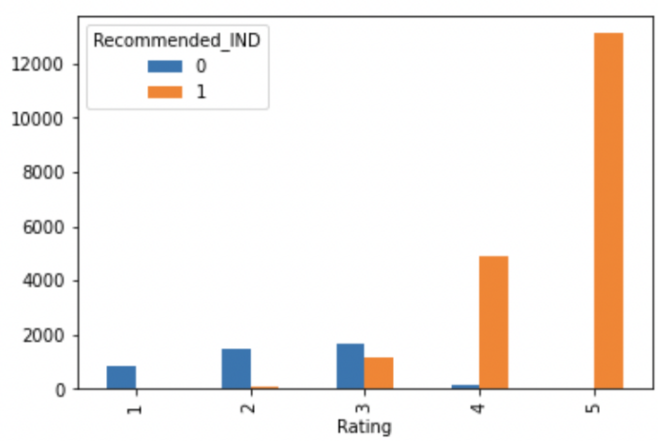
Below are the bar charts of “Division_Name”, “Class_Name”, “Department_Name”, and “Rating" by “Recommended_IND”.

```
data_train.groupby(['Division_Name','Recommended_IND'])['Clothing_ID'].count().unst
ack('Recommended_IND').plot.bar()
data_train.groupby(['Class_Name','Recommended_IND'])['Clothing_ID'].count().unstack
('Recommended_IND').plot.bar()
data_train.groupby(['Department_Name','Recommended_IND'])['Clothing_ID'].count().un
stack('Recommended_IND').plot.bar()
data_train.groupby(['Rating','Recommended_IND'])['Clothing_ID'].count().unstack('Re
commended_IND').plot.bar()
```









3. Methodology

Sentiment Analysis

Next, we apply VADER for our sentiment analysis. We use compound score as the main scoring for our text.

```
# Define VADER
vader = SentimentIntensityAnalyzer()

def detect_tb_polarity(text):
    return TextBlob(text).sentiment.polarity

def detect_vader_comp(text):
    return vader.polarity_scores(text)['compound']

# Calculate Compound Score
data_train_1 = data_train.copy()
data_train_1['tb_polarity'] = data_train_1.Review_Text.apply(detect_tb_polarity)

vader = SentimentIntensityAnalyzer()
data_train_1['vader_comp'] = data_train_1.Review_Text.apply(detect_vader_comp)
```

After calculation, we got a new dataframe as below.

```
data_train_1.head(5)
```

	Clothing_ID	Age	Title	Review_Text	Rating	Recommended_IND	Positive_Feedback_Count	Division_Name	Department_Name	Class_Name	tb_polarity	vader_comp
0	767	33	NaN	absolutely wonderful silky sexy comfortable	4	1	0	Initmates	Intimate	Intimates	0.633333	0.8991
1	1080	34	NaN	love dress s sooo pretty happened find store m...	5	1	4	General	Dresses	Dresses	0.318750	0.9700
2	1077	60	Some major design flaws	high hopes dress really wanted work initially ...	3	0	0	General	Dresses	Dresses	0.082300	0.9062
3	1049	50	My favorite buy!	love love love jumpsuit s fun flirty fabulous ...	5	1	0	General Petite	Bottoms	Pants	0.500000	0.9464
4	847	47	Flattering shirt	shirt flattering due adjustable front tie perf...	5	1	6	General	Tops	Blouses	0.458333	0.9062

Then we set 0.05 and -0.05 as our thresholds to categorize the texts. We assign text with compound score larger than 0.05 as positive text, compound score smaller than -0.05 as negative text, others as neutral text.

```
scores = data_train_1.vader_comp

Label = []
length = len(scores)
for i in range(length):
    Score = scores[i]
    if Score >= 0.05: # Positive Text
        Label.append(1)
    elif Score <= -0.05: # Negative Text
```

```
        Label.append(0)
    else:
        Label.append(2) # Neutral Text
```

Next, we drop the neutral text and get a new dataframe with only the Review_Text and Label.

```
data_last.drop(data_last[data_last['Label']==3].index, inplace=True)
data_last.head()
```

	Review_Text	Label
0	absolutely wonderful silky sexy comfortable	1
1	love dress s sooo pretty happened find store m...	1
2	high hopes dress really wanted work initially ...	1
3	love love love jumpsuit s fun flirty fabulous ...	1
4	shirt flattering due adjustable front tie perf...	1

Lastly, we update stopwords with some frequently appeared words, and then produce word clouds for the positive and negative reviews.

```
# Update Stopwords
stopwords.update(['nan', 'dress', 'shirt', 's', 'm', 'top', 'color', 'bought',
'look', 'ordered'])

# Word Cloud
Positive_Data = data_last[data_last["Label"] == 1]
Positive_Data = Positive_Data["Review_Text"]
Negative_Data = data_last[data_last["Label"] == 0]
Negative_Data = Negative_Data["Review_Text"]

# Convert Data to String
Positive_Data_2 = Positive_Data.to_string()
Negative_Data_2 = Negative_Data.to_string()

print("Positive words are as follows")
Wordcloud_P = WordCloud(stopwords=stopwords,
background_color='white').generate(Positive_Data_2)
plt.figure(figsize=(10,10))
plt.imshow(Wordcloud_P, interpolation='bilinear')
plt.axis("off")
plt.show()


print("Negative words are as follows")
Wordcloud_N = WordCloud(stopwords=stopwords,
background_color='white').generate(Negative_Data_2)
plt.figure(figsize=(10,10))
```

```
plt.imshow(Wordcloud_N, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Word Cloud - Positive reviews

Word Cloud - Negative reviews

Predictive Modeling

Because there are 15152 positive rows and 485 negative rows, the dataset is imbalanced. To solve this problem, we used SMOTE to oversample the data in the “negative” class. After the process, the amount of both two classes are 15152 rows.

```
sm = SMOTE(random_state=2)
X_train_sm, Y_train_sm = sm.fit_resample(X_train, Y_train)
```

Then, we built 4 models and made predictions. The 4 models are: Logistic Regression, Naive Bayes, Decision Tree and Random Forest.

To compare the performance of prediction of the models, we calculated Accuracy, Recall, F1-Score, Confusion Matrix, ROC and AUC for every model.

```
models = [
{
    'label': 'Logistic Regression',
    'model': LogisticRegression(random_state=0),
},
{
    'label': 'Naive Bayes',
    'model': MultinomialNB(),
},
{
    'label': 'Decision Tree',
    'model': DecisionTreeClassifier(),
},
{
    'label': 'Random Forest',
    'model': RandomForestClassifier(max_depth=13, random_state=0),
}
]

for n in models:
    model = n['model']
    model.fit(X_train_sm,Y_train_sm)
    Y_pred=model.predict(X_test)

    print('%s'%n['label'])
    print("Accuracy: "+str(accuracy_score(Y_test, Y_pred)))
    print("Recall: "+str(recall_score(Y_test, Y_pred, average="weighted")))
    print("F1-Score: "+str(f1_score(Y_test, Y_pred, average="weighted")))
    print("Confusion Matrix")
    print(confusion_matrix(Y_test, Y_pred))
    print(classification_report(Y_test, Y_pred))

plt.figure(0).clf()

for n in models:
    model = n['model']
    model.fit(X_train_sm,Y_train_sm)
    Y_pred=model.predict(X_test)

   fpr, tpr, thresholds = metrics.roc_curve(Y_test,
model.predict_proba(X_test)[:,1])
```

```
    auc = metrics.roc_auc_score(Y_test, model.predict(X_test))

    plt.plot(fpr, tpr, label='%s ROC (area = %.2f)' % (n['label'], auc))

plt.legend()
```

Performance of Models

Model	Accuracy	Recall	F1-Score	AUC
Logistic Regression	**0.9575**	**0.9575**	**0.962**	**0.82**
Naive Bayes	0.9205	0.9205	0.9363	0.76
Decision Tree	0.9421	0.9421	0.9473	0.68
Random Forest	0.9550	0.9550	0.9535	0.63

Confusion Matrix

Logistic Regression

	P	N
P	157	75
N	210	6263

Naive Bayes

	P	N
P	137	95
N	438	6035

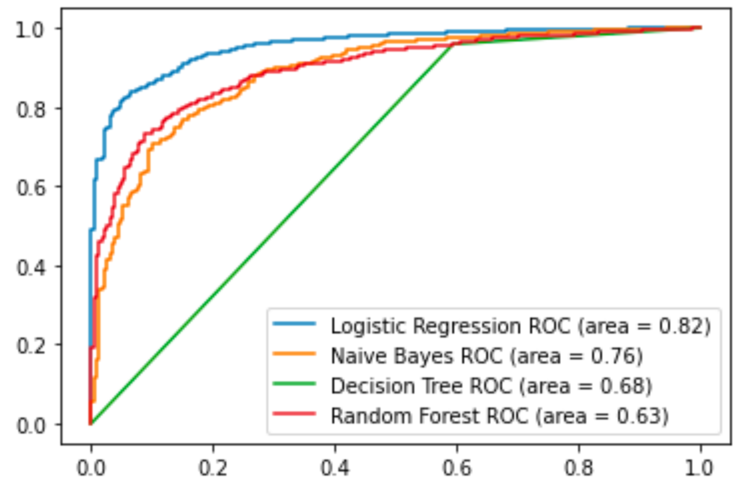
Decision Tree

	P	N
P	91	141
N	247	6226

Random Forest

	P	N
P	66	166
N	136	6337

ROC

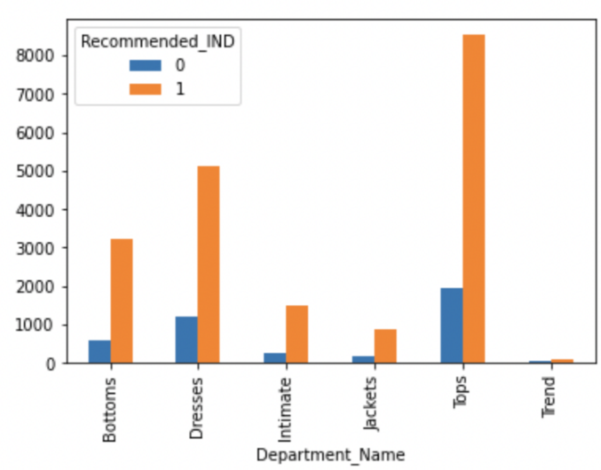


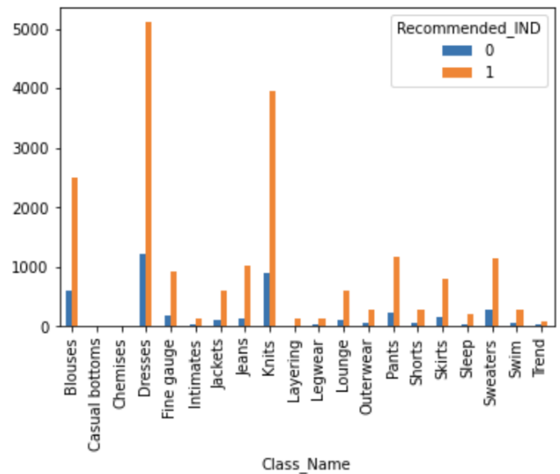
Logistic Regression is the best model. Its F1 Score and AUC are both the highest among the four models.

4. Key findings

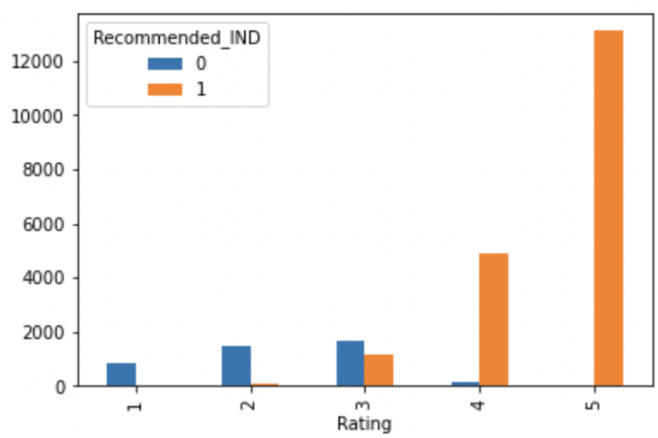
Based on the bar chart of the division name below, the general performance of the women's e-commerce platform is significant in terms of the overall recommended rate being higher than the unrecommended one. Moreover, the sold units are decreasing as the sizes of the clothes get smaller, the division in “General” has the highest sold units, “General Petite” ranks second, and “Intimates” has the lowest unit sold. In other words, compared to other different divisions, a larger proportion of customers recommend “General” than “General Petite” and “Intimate”. Based on these metrics, we should design the products in larger sizes for the following seasons, such as medium, large and extra-large. The women's e-commerce platform has to embrace the new trend of body positivity, though it could be a sharp turn away from the styles that defined the women's apparel industry for decades. There will be a more promising future for the e-commerce platform when it provides a wider variety of types of clothes.

Product quality consistency is a principle to the overall success of every business. Providing consistency allows customers to know what to expect every time they merchandise and every product they purchase, which could increase the trust of the customers towards the brand and increase sales units in the long term. Based on "Class name" and "Department name", we can find out the unrecommended rate in Blouses, Dresses, Knits, and Top are relatively higher than in other categories. Customers are able to observe and realize the quality of clothing products. By improving the product consistency aim for these four types of segments could be a significant increase to the recommended rate.





According to the rating bar chart below, we could tell the rating score (from 1 to 5) is positively related to the customers' recommended propensity. The compulsive takeaway is where we could look through the customers who rate the platform less than 3 stars, their recommendation inclination is inconsistent. So, the platform could enforce the customer relations management with this segment of customers, which could potentially turn their negative shopping experience into a positive one. For the customers who rate the platform at 4 or 5, product consistency is a key to maintaining customer retention; Also, for the customers who rate the platform under 3, which could be the churn rate that is a loss for the platform. In addition, the proportion of ratings more than 3 stars on the condition of customers who recommend (81%) is higher than ratings less than 3 stars on the condition of customers who don’t recommend (19%).



Wordcloud

Fabric and Sweaters are the words that show up more frequently in both positive and negative comments. Based on that, the platform should narrow the customer segmentation, to clarify which types of customers could be the target audience depending on the type of fabric they choose. Moreover, if the platform would like to expand the business: create multiple product lines, to target different types of customers, such as a high-end line for customers at the age of 28-35, with better financial capability and a classic line for the customer at the age of 18-25, with relatively less money to spend on apparel.

Pants are the category of products that have been mentioned significantly in the positive comments. Based on the metrics, the women's e-commerce platform could take the product categories as advertised to attract new customers based on the high customer reputation.

Size is the biggest concern of customers, which could be related to the bar chart of "division name". The women's e-commerce platform produces mainly smaller sizes for the customers, which could harm the platform's reputation. As our team has mentioned before, the platform could benefit from producing the medium, large and extra-large sizes of apparels.

5. Challenges

During the journey of working on our projects, we overcame two main challenges below. Firstly, we found that the dataset was imbalanced between the recommended and not recommended comments. In order to solve the problem, we used SMOTE to oversample the data in the “negative” class. After the oversampling, we were able to build further different models, and compared and evaluated their performances.

Second, we also encountered the challenge of how we evaluate different models as a consistent benchmark. Due to the advantages of the ROC curve, we decide to use it as a criterion to indicate the performance of multiple models at all classification thresholds. With higher AUC scores, which compound score is higher than 0.5, we are able to measure the better performances of the models.

Second, to optimize the model Decision Tree, we used Grid Search and Randomized Search for the optimal parameters for the model. However, the accuracy rate and auc was lower than the result from the default parameters. The reason may be related to the range of the value of parameters we chose for the process of Grid Search and Randomized Search.

6. Conclusion

Performance of Models

Model	Accuracy	Recall	F1-Score	AUC
Logistic Regression	**0.9575**	**0.9575**	**0.962**	**0.82**
Naive Bayes	0.9205	0.9205	0.9363	0.76
Decision Tree	0.9421	0.9421	0.9473	0.68
Random Forest	0.9550	0.9550	0.9535	0.63

Compared to the different performance of each model we used, the model using Logistic Regression performs the best. Because its AUC score and F1-Score are the highest, 0.82 and 0.96 among all.

From those charts and tables presented above, we conclude the following points. Firstly, the women's e-commerce platform can expand the amount of product by producing larger sizes of apparels. Moreover, customers do value the importance of product quality, which reflects on higher ratings and more positive reviews. The platform should make more efforts to improve and maintain its consistent product quality so that the platform can attract new potential customers. Hopefully, as long as the platform is willing to adjust and adapt to the trends nowadays, it leads to higher profits, better reputation and loyalty to the platform.