Capstone Project 2 -Predicting Women’s Clothing Ratings

Final Report

Marisa Mitchell

Introduction

This project investigates the ratings of women’s clothing items. The potential client for this project is a women’s fashion and clothing e-commerce company. This company cares about this problem because they want to receive positive reviews of their clothing items which will help them in a variety of ways. First, positive reviews indicate items that consumers are happy with. A fashion company would want consumers to be happy with their product because they are more likely to recommend these products and to purchase again from the company. Additionally, positive reviews are helpful to the company as they are visible on their website for potential new clients to view and will shape the decision made by the potential client. Finally, a company could use information about the negative reviews to potentially improve an item. For instance, a negative review may state that the customer did not like the fabric a particular clothing item is made with. The company could then use this information to change the fabric that item is made with to be more appealing to their clients. In this project I aim to determine factors in the review influence the rating of women’s clothing items.

Dataset

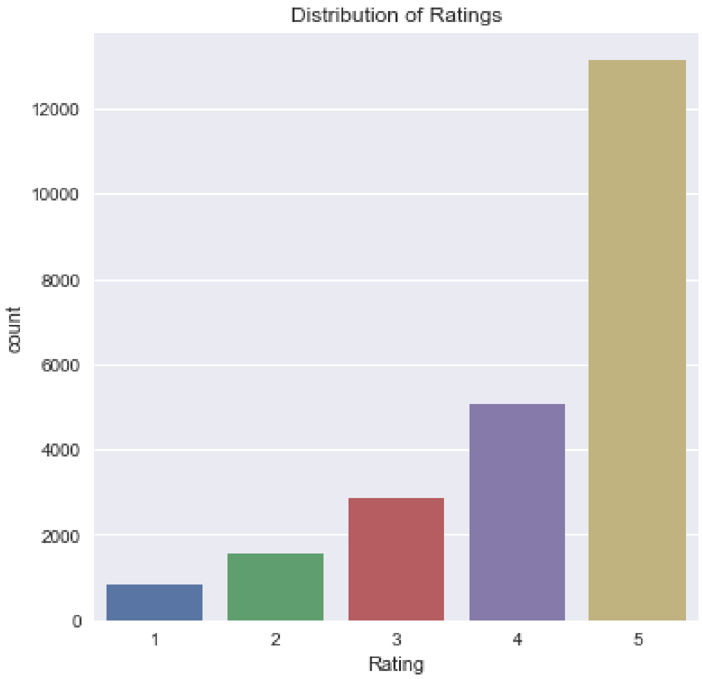
For this project I used the Women’s E-Commerce Clothing Reviews Dataset from Kaggle located at https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews. This dataset contained 10 variables related to 23486 clothing reviews. The variables in the dataset were clothing ID, the age of the reviewer, the title of the review, the review text, the rating of the item by the reviewer, an indicator of whether the product is recommended by the reviewer, the number of positive feedback counts on the review, the name of the division the product is in, the name of the department the product is in, and the product class. The clothing ID was a numeric variable identifying the reviewer. The ages were continuous and ranged from 18 to 99 with a average age of 43. The title of the review and review text were fields that contained the narrative of the review and a brief title of the review. The ratings were on a scale of 1 to 5 with an average item rating of 4. The recommended indicator was a dichotomous variable with values of 0 or 1. Division name, department name, and class name were all categorical variables which described the division, department, and class of the item. The dataset had 3,210 missing values for the title, 845 missing values for review text, and 14 missing values for division name, department name, and class name. I created a new variable merging the text from the title and the review text which I called "title_review". This new variable was missing 844 values. I also created a new variable in the dataset which categorized the ratings into three groups: high (ratings of 4 and 5), medium (ratings of 3), and low (ratings of 1 and 2).

Potential other datasets that could be used would be internal reviews from a women's ecommerce company who is interested in examining their own items.

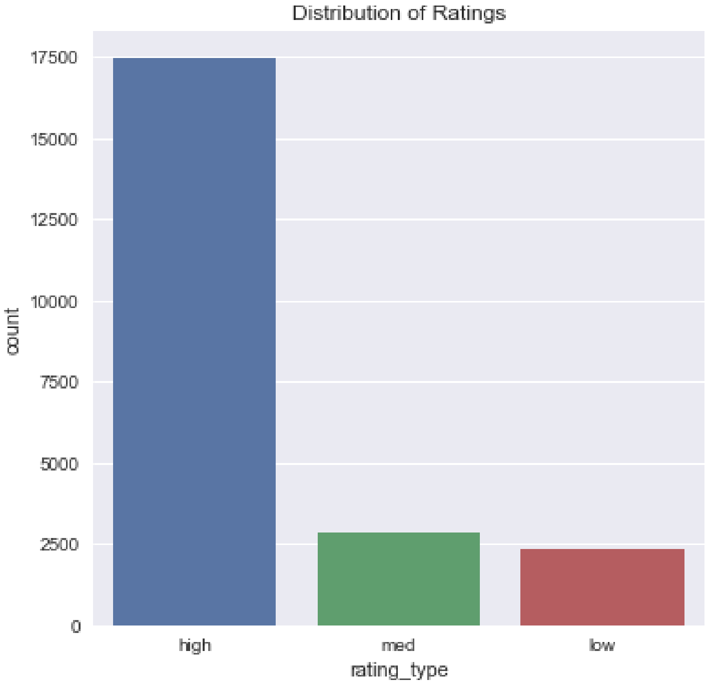
Exploratory Analysis

The findings of this report are given without the complete Python code. The Python code in which the findings are based can be found on Github.

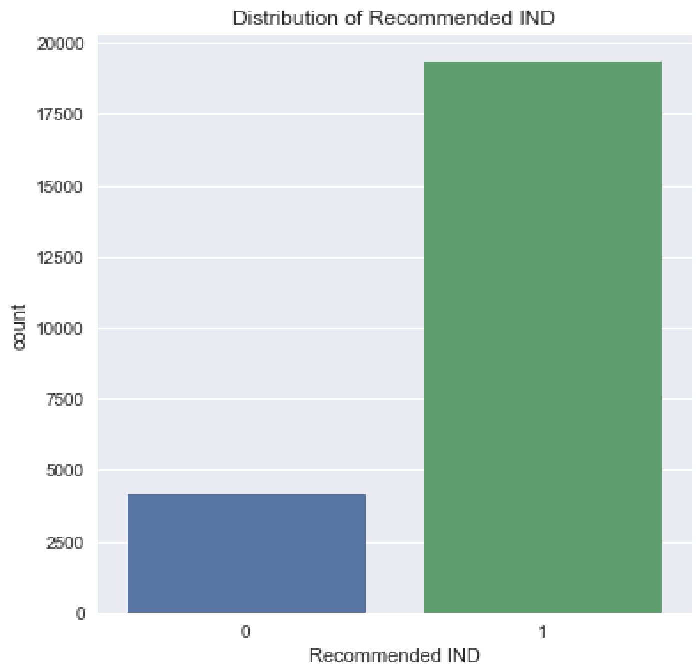
Before developing any predictive models I conducted exploratory data analysis to examine the data and conduct visualization which might point out patterns in the data.



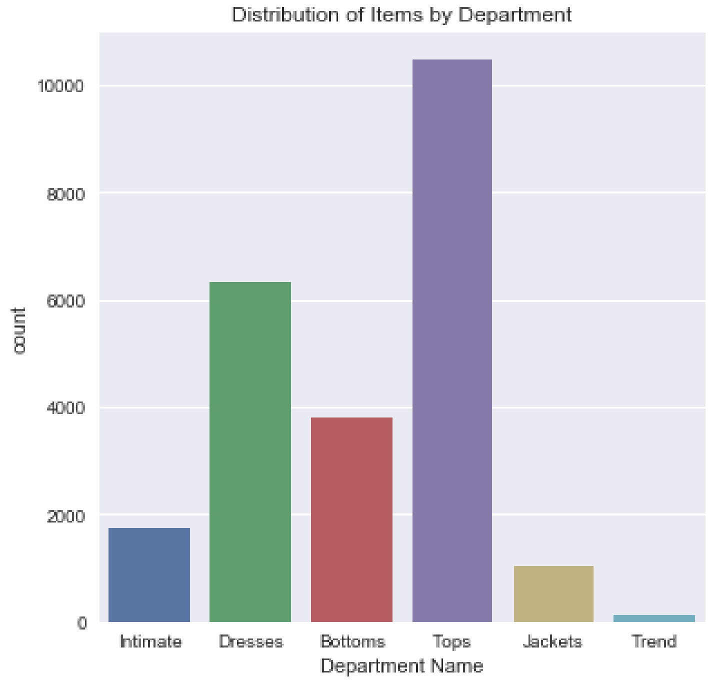
The above bar graph illustrated that the ratings were skewed in the positive direction with most ratings being 4 or 5 indicating satisfaction with the item.



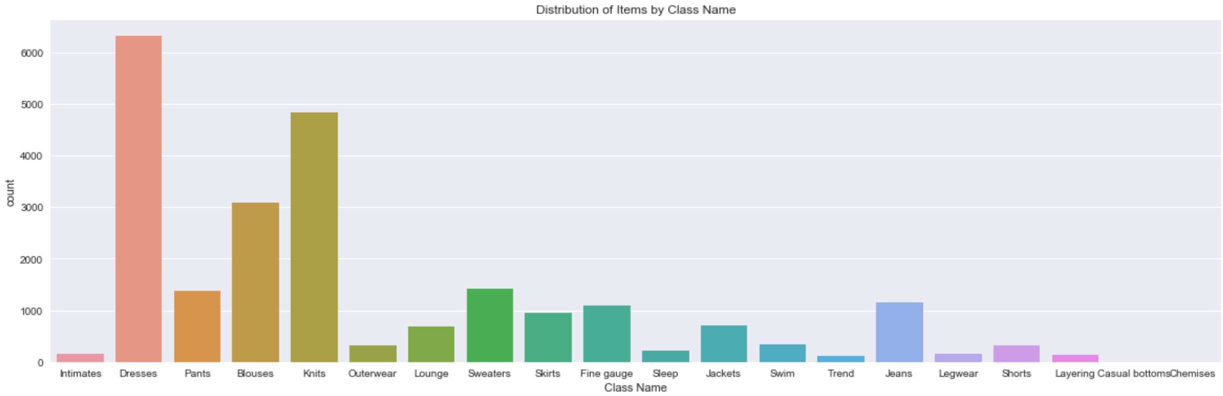
The above bar graph shows the distribution of ratings once the data was binned. This shows that the high ratings were most frequent and the low ratings were the least frequent. The data is shown to be unbalanced between the three categories.



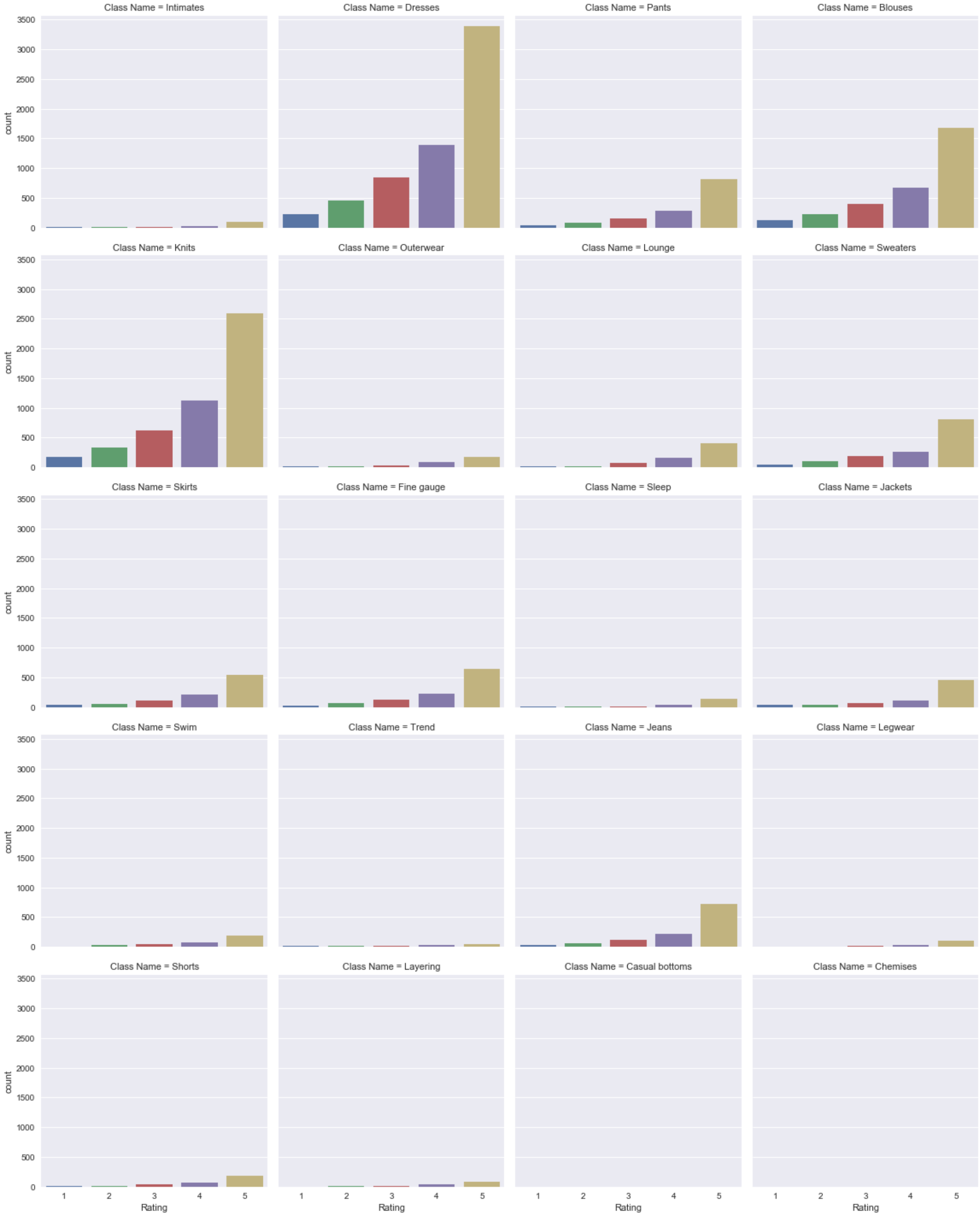
The above bar chart illustrates that the majority of reviewers would recommend the item.



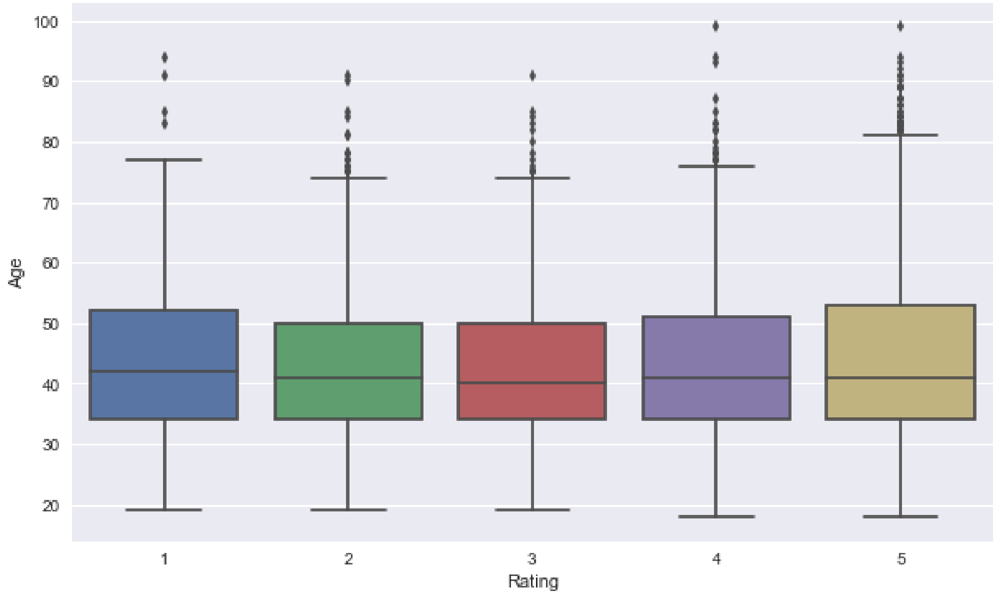
The above bar chart illustrates that the most frequent department type was tops. The next highest item type was dresses followed by bottoms, intimates, and jackets. The items in the trend department were the least frequent in this dataset.

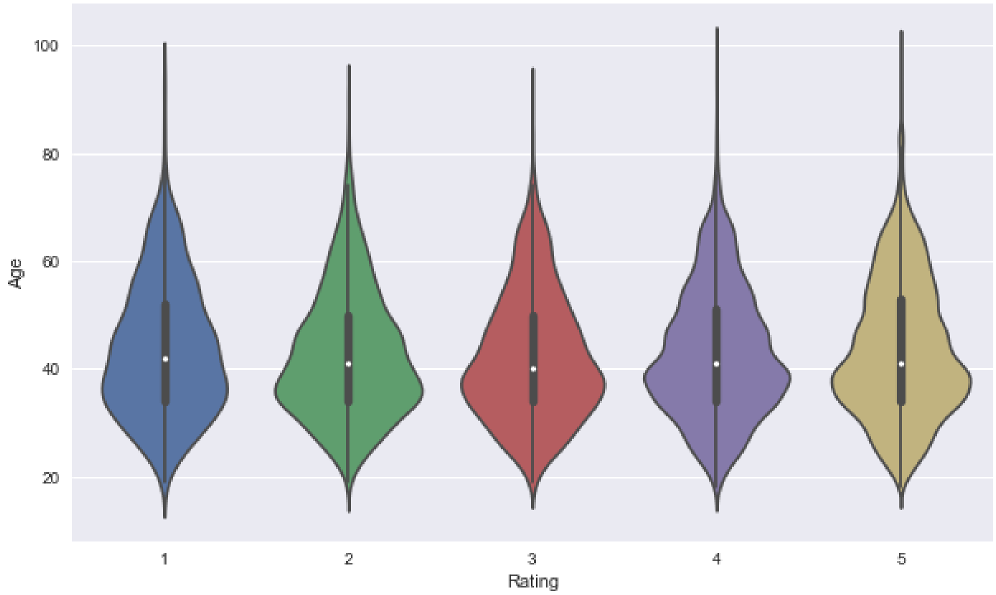


The above bar chart further visualizes the types of items in the dataset by looking at the frequency of each class name. Here it appears that dresses, blouses, and knits were the most frequent items while trend, casual bottoms, and chemises are the least frequent items in the dataset.

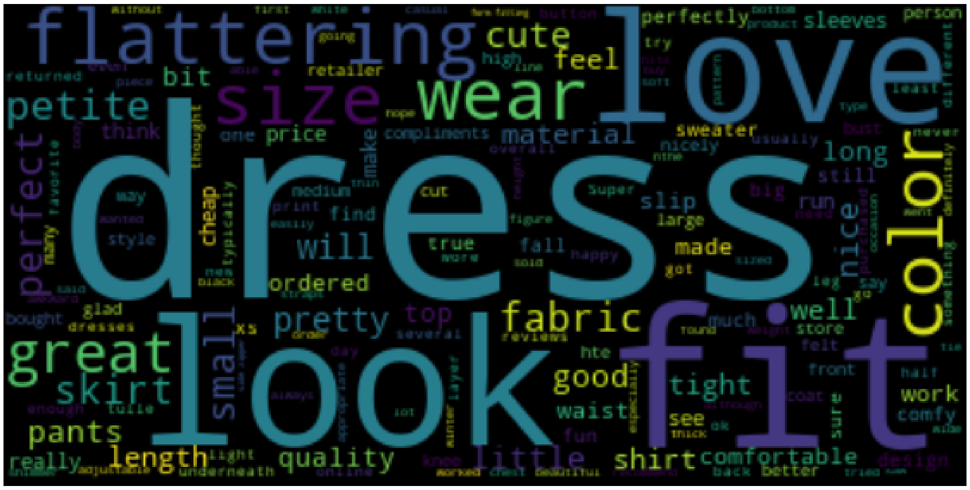


The above bar graphs show the distribution of rating for each class name. This illustrates that across the class names the distribution of ratings is approximately the same with a rating of 1 always being the least frequent and then increasing to rating 5 which is the most frequent.

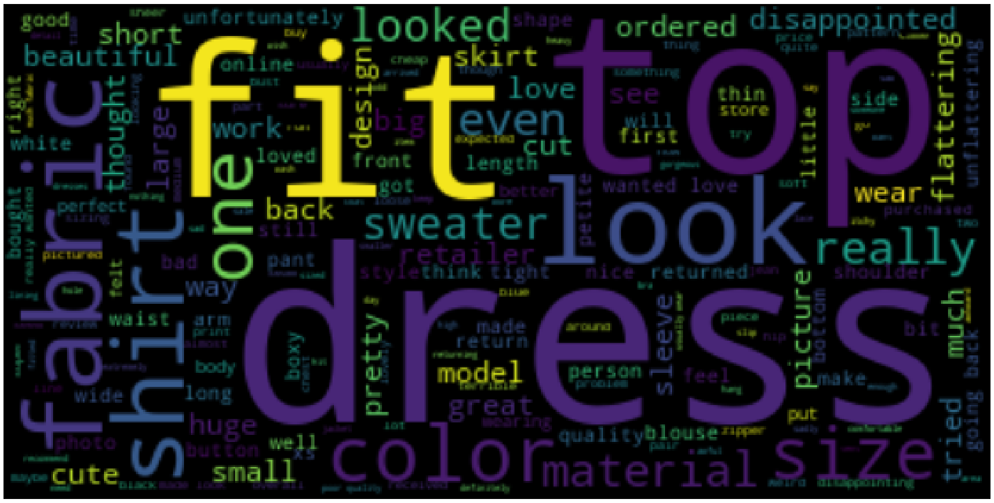




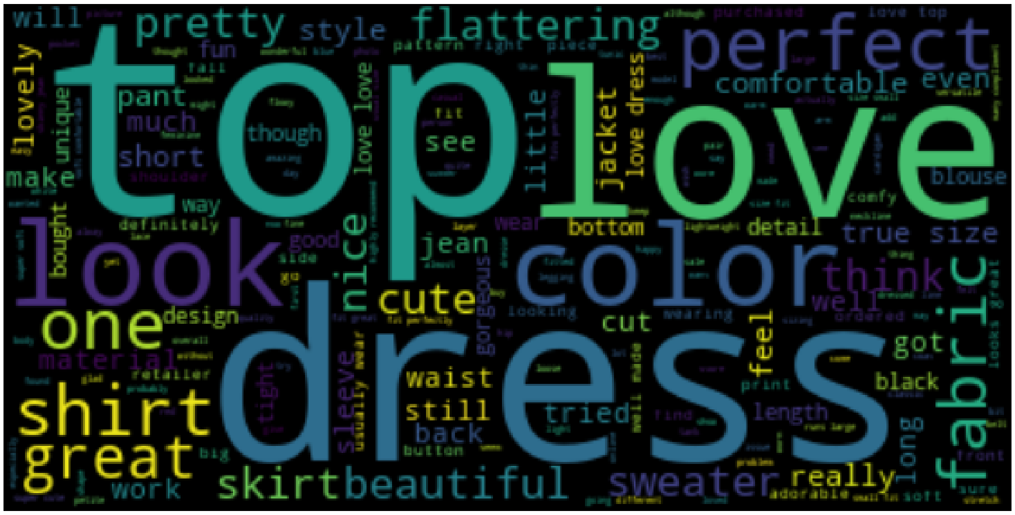
The above box and violin plots illustrate that the distribution of age across all the ratings was approximately the same.



The above word cloud was created using the text from the title_review variable across all reviews. The word cloud shows that the words dress, look, fit, love, flattering, size, wear, and color all appeared frequently across all reviews.



The above word cloud was developed using the text from the title_review variable for all the reviews that were categorized as low. This word cloud shows a high frequency of the words, fit, top, dress, fabric, shirt, and look.



This final word cloud was developed using the text from the title review variable for reviews which the rating was categorized as high. The high frequency words in these reviews are top, dress, look, love, and perfect.

Text Pre-Processing

In order to prepare text for use in machine learning I conducted several text pre-processing steps with the title_review variable I created. The first step I took was removing accented characters so that they are converted to normal English charters . This process helps standardize the words for analysis. Next, I took the title_review text and expanded contractions to further standardize the text. This process converted any instance of a contractration to the two words that are contained in the contraction (e.g., *don't* would be changed to *do not*). Following expanding contractions I removed any special characters or symbols that were not alphanumeric. The next pre-processing step I took was stemming which is a process that removes inflections from words and keeps the base or root word instead. This process would take the words *complemented* and *complementing* and convert them to *complement*. Next, I used lemmatization to remove affixes from the base word. The final step in text pre-processing that I took was the remove stopwords. This process removes common words that have little significance to the meaning of text such *as a, an, the*, and *or*.

Feature Engineering

Once the pre-processing of the title_review text was done I was able to begin converting this to features to be used in my machine learning models. I chose to engineer the text using two different vectorizers. The first vectorizer was the count vectorizer which creates a sparse matrix in which each word in the corpus is a separate features and each row holds a raw frequency count of each word that is contained in each document. This method is often called bag or words. The other method I used was the Term Frequency-Inverse Document Frequency (TF-IDF) method of vectorization. This method creates a sparse matrix for each document but scales and normalizes the values in the matrix by using the following calculation:

tf-idf(w, d) = bow(w, d) * N / (# documents in which word w appears)

The TF-IDF function creates values of close to 1 for words that appear in many documents and larger values and higher values for those words that appear in few documents.

Analytical Method

The objective of this project was the produce an algorithm that would optimize classification of high, medium, and low clothing ratings based on the text in the title and review of the item. I used 6 classification algorithms using each set of features (the bag of words features and the TF-IDF features) and used the binned categories of ratings as the labels. The 6 algorithms I used were KNN, logistic regression, linear support vector machine, decision tree, random forest, and gradient boosting. This yielded a total of 12 default models that I ran. For each model I examined the accuracy score, the confusion matrix, and the classification report. Using the results of the default models I chose the top 3 models and performed additional hyperparameter tuning.

Model Results and Evaluation

Initial Results

All models were run prior to hyper-parameter tuning using default settings. Results are shown in the below table. I used test data accuracy scores to select the highest performing models. This score represents the percent of cases that the model predicted the correct label. In this case the number of reviews that were correctly labeled with their actual rating group of high, medium, or low. There were three models which had the highest accuracy. Those models were the bag of words logistic regression model, the TF-IDF logistic regression model, and the TF-IDF linear SVM model. The accuracy on the test data for these three models was .83 indicating that 83% of the ratings were correctly predicted based on the text in the title_review. The decision tree models in both bag of words and TF-IDF models were the lowest performing with an accuracy of .74.

Model	Accuracy Before Tuning
Bag of Words	
KNN	0.77
Logistic Regression	0.83
Linear SVM	0.80
Decision Tree	0.74
Random Forest	0.79
Gradient Boosting	0.80
TF-IDF	
KNN	0.79
Logistic Regression	0.83
Linear SVM	0.83
Decision Tree	0.74
Random Forest	0.79
Gradient Boosting	0.81

Model Tuning Results

In order to tune the hyper-parameters of the three models tied for highest performing I used the grid search function in the scikit-learn package. For both the bag of words and the TF-IDF logistic regression models I tuned on penalty and C. The penalty values I used were l1 and l2. I used the values of .001, .01. 1, 10, and 100 for the C values. For the linear SVM model I tuned on the class weight and C values. For class weight I used balanced and none. For C values I used .001, .01, 1, 10, 100, and 1000. Overall accuracy results are shown in the below table.

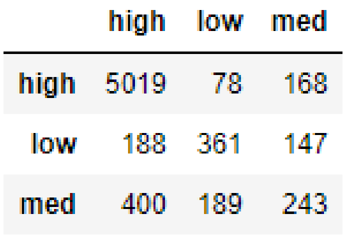
Model	Accuracy Before Tuning	Accuracy After Tuning
Bag of Words		
Logistic Regression	0.83	0.83
TF-IDF		
Logistic Regression	0.83	0.83
Linear SVM	0.83	0.83

Overall accuracy of the three models did not increase after tuning the hyperparameters. All three models maintained an accuracy score of .83.

To further evaluate these models I examined the confusion matrix and classification report for each of these models using the best estimators. Below you will find tables of the confusion matrix and classification report for each of these models.

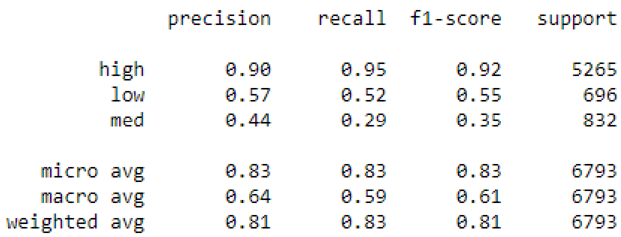
Bag of Words Logistic Regression

Confusion Matrix



	high	low	med
high	5019	78	168
low	188	361	147
med	400	189	243

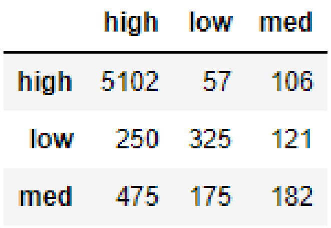
Classification Report



```
              precision    recall  f1-score   support

        high       0.90      0.95      0.92      5265
         low       0.57      0.52      0.55       696
         med       0.44      0.29      0.35       832

   micro avg       0.83      0.83      0.83      6793
   macro avg       0.64      0.59      0.61      6793
weighted avg       0.81      0.83      0.81      6793
```

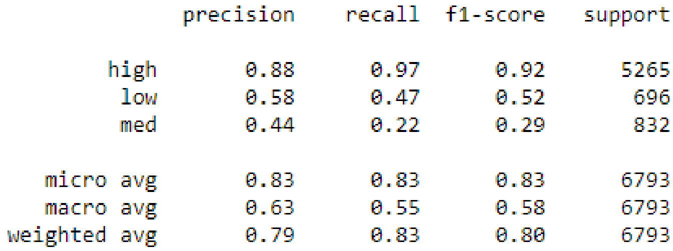
TF-IDF Logistic Regression

Confusion Matrix



	high	low	med
high	5102	57	106
low	250	325	121
med	475	175	182

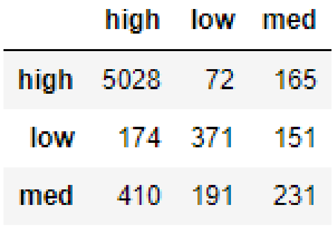
Classification Report



```
              precision    recall  f1-score   support

        high       0.88      0.97      0.92      5265
         low       0.58      0.47      0.52       696
         med       0.44      0.22      0.29       832

   micro avg       0.83      0.83      0.83      6793
   macro avg       0.63      0.55      0.58      6793
weighted avg       0.79      0.83      0.80      6793
```

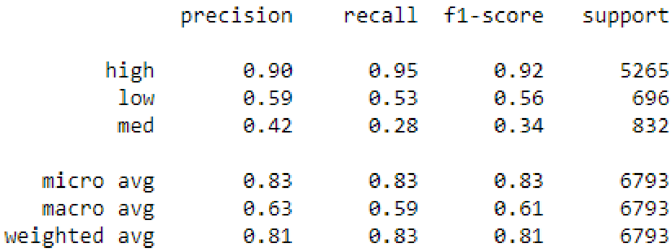
TF-IDF Linear SVM

Confusion Matrix



	high	low	med
high	5028	72	165
low	174	371	151
med	410	191	231

Classification Report



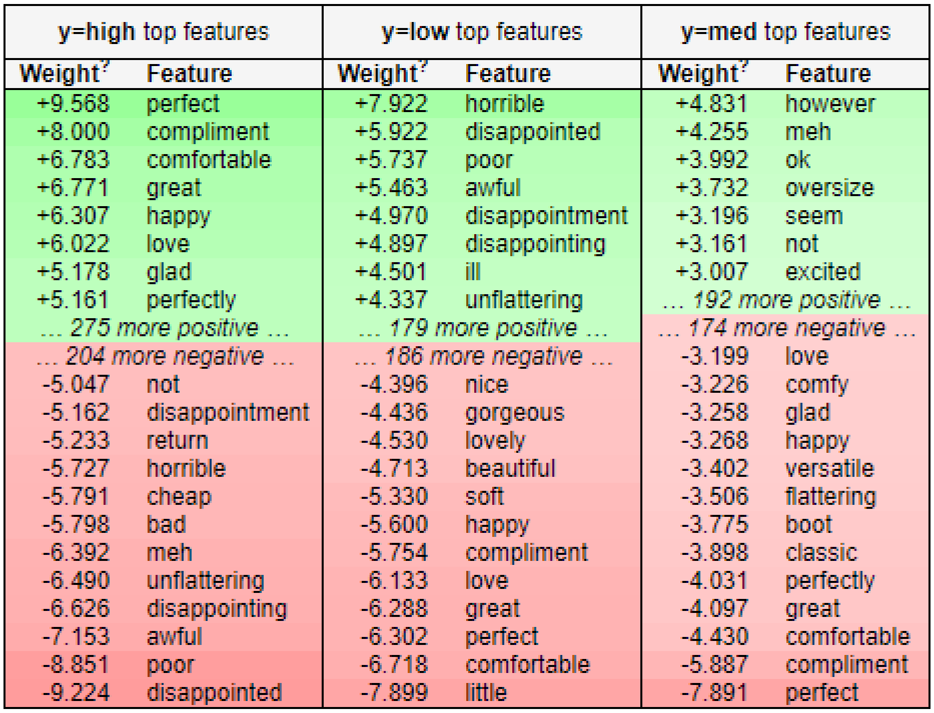
```
              precision    recall  f1-score   support

        high       0.90      0.95      0.92      5265
         low       0.59      0.53      0.56       696
         med       0.42      0.28      0.34       832

   micro avg       0.83      0.83      0.83      6793
   macro avg       0.63      0.59      0.61      6793
weighted avg       0.81      0.83      0.81      6793
```

All three of the models had similar results in their confusion matrix and classification report with the highest levels of precision(percentage of items identified in a group were actually in the group) and recall (percentage of items in a group were correctly identified by the model).

High precision and recall scores (between .88 and .97) were calculated for the reviews that received high ratings. All three models had precision and recall scores at approximately .50 for the reviews in the medium ratings group. The precision scores were approximately .40 for the reviews in the low ratings group. The recall scores were between .22 and .29 for the reviews in the low ratings group.

Conclusions

The purpose of this project was the create a model which was able to use the text from reviews of clothing items to predict the rating that item received. This would help e-commerce fashion companies understand why items received ratings and potentially make adjustments to their product offerings. Due to the similar results of the bag of words logistic regression, TF-IDF logistic regression, and TF-IDF linear SVM models any one of them could be used for this purpose. I would recommend the TF-IDF logistic regression model as the logistic regression model is easy to interpret and the TF-IDF process builds upon the bag of words method by emphasizing how important a word is to a particular document.



y=high top features		y=low top features		y=med top features	
Weight?	Feature	Weight?	Feature	Weight?	Feature
+9.568	perfect	+7.922	horrible	+4.831	however
+8.000	compliment	+5.922	disappointed	+4.255	meh
+6.783	comfortable	+5.737	poor	+3.992	ok
+6.771	great	+5.463	awful	+3.732	oversize
+6.307	happy	+4.970	disappointment	+3.196	seem
+6.022	love	+4.897	disappointing	+3.161	not
+5.178	glad	+4.501	ill	+3.007	excited
+5.161	perfectly	+4.337	unflattering	*… 192 more positive …*	
… 275 more positive …		*… 179 more positive …*		*… 174 more negative …*	
… 204 more negative …		*… 186 more negative …*		-3.199	love
-5.047	not	-4.396	nice	-3.226	comfy
-5.162	disappointment	-4.436	gorgeous	-3.258	glad
-5.233	return	-4.530	lovely	-3.268	happy
-5.727	horrible	-4.713	beautiful	-3.402	versatile
-5.791	cheap	-5.330	soft	-3.506	flattering
-5.798	bad	-5.600	happy	-3.775	boot
-6.392	meh	-5.754	compliment	-3.898	classic
-6.490	unflattering	-6.133	love	-4.031	perfectly
-6.626	disappointing	-6.288	great	-4.097	great
-7.153	awful	-6.302	perfect	-4.430	comfortable
-8.851	poor	-6.718	comfortable	-5.887	compliment
-9.224	disappointed	-7.899	little	-7.891	perfect

The above figure shows the top 20 words from the TF-IDF model that are positively and negatively associated with reviews which received high, medium, and low ratings. According to this model high ratings are associated with being comfortable, getting compliments, and customers being glad and happy while not being associated with returns, being cheap, unflattering, or poor. Medium ratings are associated with being oversized and ok while not being associated with words like love comfy, glad, happy, and flattering. Finally low ratings are associated with disappointment, unflattering, poor, and horrible while not being associated with being nice, gorgeous, beautiful, soft, complement, and comfortable.

Next Steps

One next step of this analysis could be to examine if oversampling or undersampling techniques might help with model accuracy due to the unbalanced item ratings that were observed in this dataset.   Additionally, other variables in the dataset could have been added as features in the models to see if that had any potential impact on the accuracy of the prediction of item ratings. Further, it may help to examine feature weights for each individual clothing category to see if there are words associated with the clothing category. These weights could help the e-commerce company determine factors that influence ratings of particular items and then adjust those items accordingly or discontinue items based on these results. Feature weights for individual clothing ID’s could similarly help the company by examining their individual items and what they can adjust to make customers more satisfied.