Problem Formulation

Our capstone project aimed to help a fashion industry startup to improve their customer wardrobe inventory Application. The App gathers the information about the customers' wardrobe inventory from their emails and catalogs the products into 12 different product categories.

Identifying the women's products and correctly classifying them into different categories is the big challenge. Through this project, we built machine learning models for the company to efficiently and accurately predict the product category.

Data Exploration

Given a new email receipt with information such as brand Id, retailer data, item name, retailer Id, and category Id we want to assign the products on it into one of the 12 categories.

Output: Product Category

- Tops – 110
- Bottoms - 120
- Dresses - 140
- Jumpsuits - 130
- Outerwear - 150
- Activewear - 160
- Beachwear- 170
- Shoes - 200
- Bags - 300
- Accessories 400
- Beauty 500
- Miscellaneous 600
- Kids 610
- Mens 620
- None 0

The information we are interested in such as 'item name', 'brand name' and 'category Id' exists in strings. This is a supervised machine learning text classification problem. Predicting the right category on the provided string will help this company best serve its clients.

To tackle this problem, we investigated which supervised methods are best suited to handle text data, multi-class classification and imbalanced classes. Upon cleaning the data, engineering features, and balancing classes, we implemented Naive Bayes, Multinomial Logistic Regression, Support Vector Machine and Tree-based models.

The following sections walk through our process to optimize our predictions.

Imbalanced Classes

We see that the number of products per category is imbalanced (Figure 1). Kids, men's, miscellaneous, beauty and non-wardrobe items were least represented. Conventional algorithms are often biased towards the majority class, not taking the data distribution into consideration. In the worst case, minority classes are treated as outliers and ignored.

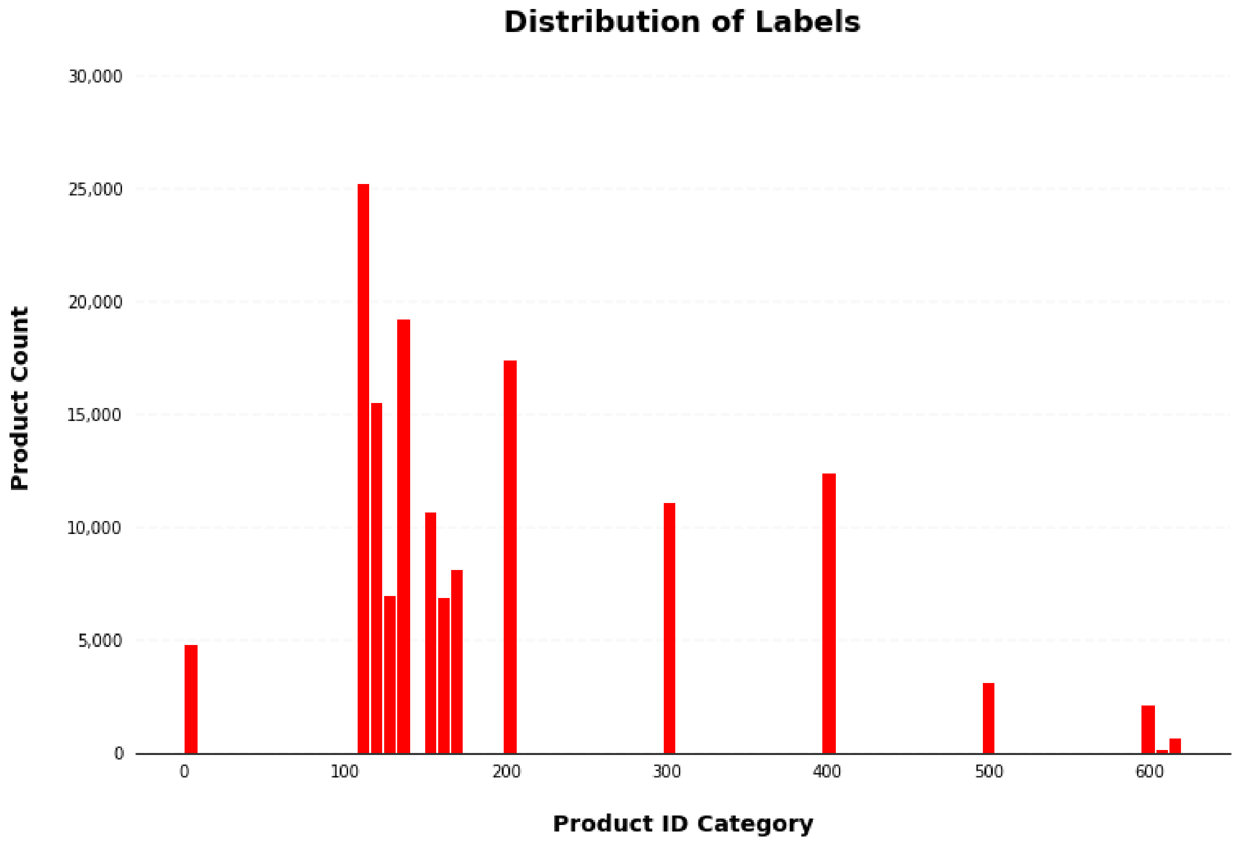


Figure 1: Distribution of the products in 12 different product categories

To overcome this problem, we undersampled the majority class, configured our models during training and also merged additional data for Men's, Kids and beauty products scraped from Flipkart and Sephora.

Feature Extraction

We combined the brand names into the product description as the product brand name appeared to be is very important to determine the product class. Products with similar item names but from different brands belonged to different categories. An item described as a ‘legging’ must belong to class 120 (bottoms) if purchased from Victoria Secret’s and must belong to class 160 (activewear) if purchased from a sporting goods brand Adidas. Hence,

The machine learning algorithms cannot directly process the text data and must be converted to numeric feature vectors. To represent the text information as numeric vectors we extracted features to use from the text as a bag of words. Using scikit learn text preprocessing tools we computed the TF-IDF vectors for each product. We tuned:

- **min_df**: the minimum numbers of documents a word must be present in to be kept.
- **ngram_range**: We trained our models on unigrams, bigrams, trigrams, four-gram and five-gram. Bigrams and trigrams performed better than unigrams.

Machine Learning and Model Evaluation

With the transformed data features and their labels, we experimented with 4 different machine learning models and evaluated their accuracy:

Logistic Regression

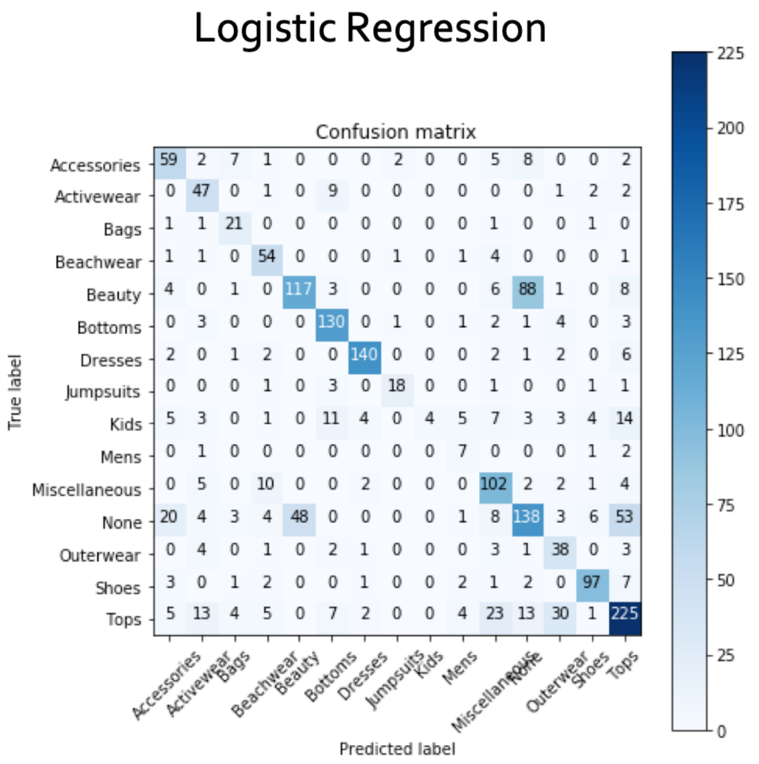
Multinomial Naïve Bayes

Support Vector Machine (Linear and Radial Kernels)

XGBoost

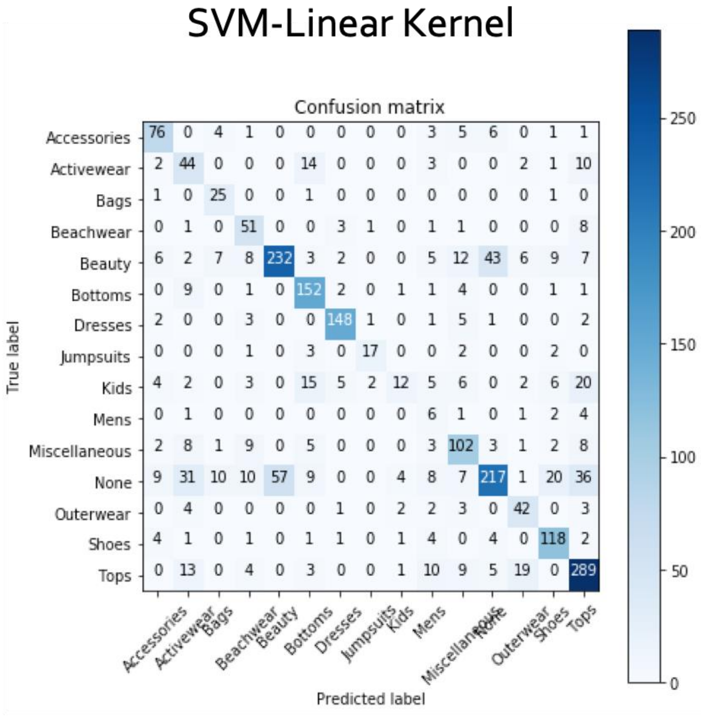
Logistic Regression and LinearSVC performed better with an accuracy of 79.44% and 77.74% compared to the other two models XGBoost (71.89%) and Naïve Bayes (70.43%).

Below are the confusion matrices from three of our models showing the discrepancies between predicted and actual labels. The vast majority of the predictions end up on the diagonal (predicted label = actual label), where we want them to be. However, there are a number of misclassifications, and it is interesting to see that these misclassified products belong the under-represented classes beauty products (class 500) and None (class 0).



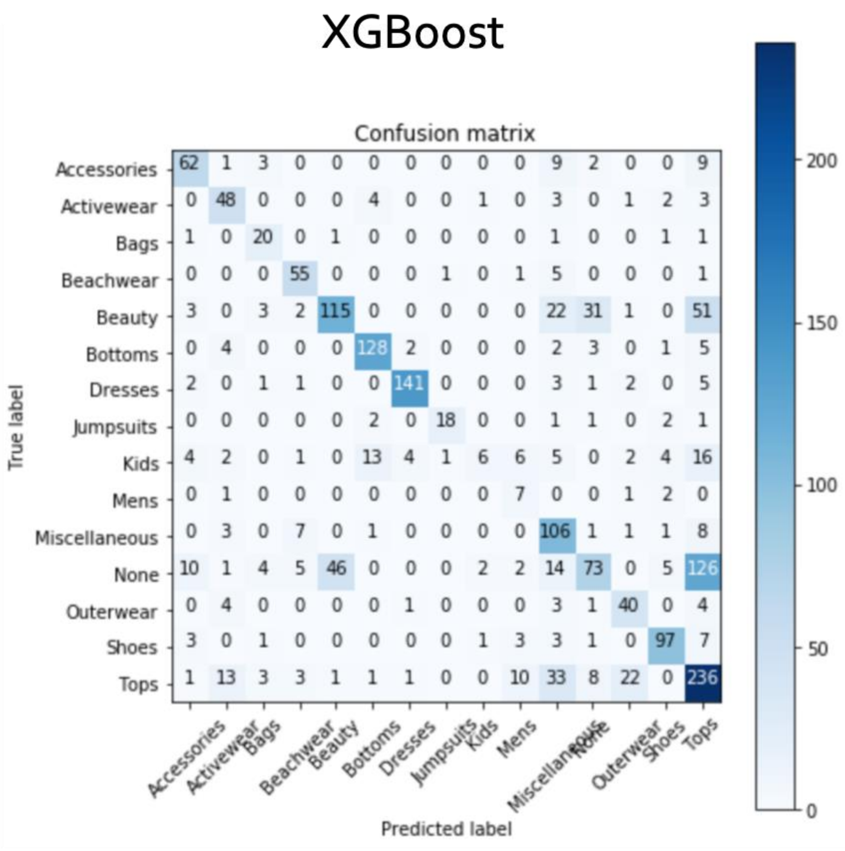
Practical Accuracy: 79.74%

Figure 2: Confusion matrix of Logistic Regression model



Practical Accuracy: 77.74%

Figure 3: Confusion matrix of SVM linear model



	Accessories	Activewear	Bags	Beachwear	Beauty	Bottoms	Dresses	Jumpsuits	Kids	Mens	Miscellaneous	None	Outerwear	Shoes	Tops
Accessories	62	1	3	0	0	0	0	0	0	0	9	2	0	0	9
Activewear	0	48	0	0	0	4	0	0	1	0	3	0	1	2	3
Bags	1	0	20	0	1	0	0	0	0	0	1	0	0	1	1
Beachwear	0	0	0	55	0	0	0	1	0	1	5	0	0	0	1
Beauty	3	0	3	2	115	0	0	0	0	0	22	31	1	0	51
Bottoms	0	4	0	0	0	128	2	0	0	0	2	3	0	1	5
Dresses	2	0	1	1	0	0	141	0	0	0	3	1	2	0	5
Jumpsuits	0	0	0	0	0	2	0	18	0	0	1	1	0	2	1
Kids	4	2	0	1	0	13	4	1	6	6	5	0	2	4	16
Mens	0	1	0	0	0	0	0	0	0	7	0	0	1	2	0
Miscellaneous	0	3	0	7	0	1	0	0	0	0	106	1	1	1	8
None	10	1	4	5	46	0	0	0	2	2	14	73	0	5	126
Outerwear	0	4	0	0	0	0	1	0	0	0	3	1	40	0	4
Shoes	3	0	1	0	0	0	0	0	1	3	3	1	0	97	7
Tops	1	13	3	3	1	1	1	0	0	10	33	8	22	0	236

Practical Accuracy: 71.89%

Figure 4: Confusion matrix of XGBoost model

Conclusions:

We achieved close to close to 80% accuracy in predicting the product class from the text data. These models can be further improved by refining our text preprocessing, gathering more information of the imbalanced classes and building an industry specific English STOPWORDS.

Code for this project can be found here.

Please contact us if you have any suggestions or questions. Thank you.