# E-commerce Product Classification

Team 32

Harshita Srinivas
Hardik Dudeja
Saksham Mehla
Mohit Bhar

INDRAPRASTHA INSTITUTE *of* INFORMATION TECHNOLOGY **DELHI**

# Motivation

## Relevance

With the growing popularity of e-commerce websites like Amazon, Flipkart and Myntra, the number of products available on these platforms has greatly increased. Owing to this, the product classification problem has gained practical significance in the industry.

## Importance

A well-built product taxonomy will allow users to easily navigate the website and explore their desired products leading to higher conversion rates and improved user experience. Appropriate product categorization will also improve search relevance on the web.

## Comparison

In contrast to manual product classification where a single product may be classified into multiple or incorrect categories by merchants, an automated classification system eliminates these inconsistencies. Thus our team was motivated to work on this real-world problem.

# Literature Review

1. Don’t Classify, Translate: Multi-Level E-Commerce Product Categorization Via Machine Translation

- This paper discusses a novel approach to the product classification task. In most proposed methods, building a well-versed product taxonomy looks at the process as a mere classification task. But the taxonomic tree is highly hierarchical with thousands of leaf categories. So the authors propose a method for visualizing the product classification task as a machine translation task.
- This approach works on utilizing the machine translation systems already in place, thereby achieving better results without incurring extra cost. It outputs the existing root to leaf paths in the product taxonomy and also outputs new paths that do not exist in the taxonomy.
- It uses novel machine transition models like the Attentional Seq2Seq model and the NMT based transformer model. Using conventional RNNs was time-consuming for the authors' approach; hence novel machine translation algorithms were utilized.
- The authors used the Rakuten dataset for the product classification task. Duplicate listings and erroneously assigned products were eliminated from the dataset for data processing, and the product titles were tokenized using an appropriate segmenter. Lastly, the above-discussed models were employed to achieve the desired results. The authors observed that the model could prune and restructure the taxonomy tree at the appropriate depth to describe the product accurately from the obtained taxonomy trees.

## 2. GoldenBullet: Automated Classification of Product Data in E-commerce

- This paper provides an insight into a software environment targeted at providing a complete solution to the product classification problem. It discusses B2B marketplaces' present popularity and the need for accurate product descriptions to have a suitable classification model.
- Golden bullet, the proposed tool developed by the authors, aims to automate every step in product classification, i.e. from information retrieval to product classification with additional UI support to manually input data.
- It uses the vector space model method for information retrieval that finds a match between the UNSPSC commodities and the product description of the data sample. Then the K-nearest neighbour and Naive Bayes classifier is implemented to search the most similar class among several categories.
- Golden bullet was implemented as a java-based client-server application. The model achieved an overall accuracy of about 80% across product descriptions in both English and French. It works at an intersection of natural language processing, information retrieval, machine learning using large volumes of manually classified data. This work can be further extended to other languages as multilingual product based classifications are necessary for real-world scenarios.

## 3. Machine Learning Based Product Classification for eCommerce

- This research paper explores classification based on users' search phrases. Here original product names and classes are supplemented with users' search phrases which improved the classification accuracy significantly.
- The research was conducted in two phases. In the first phase, product classification was done on the basis of names & classes of products. In the second phase, User generated content (search phrases) were used to classify products. At last, the results of both the phases were compared.
- The best results of text classification were  reported in experiments when linear algorithms have been applied like multiclass logistic regression, decision trees, or support vector machine (SVM) with linear kernel.
- Accuracy, Precision, Recall, and F1-score were used for quality measurement.
- It was observed that merging search phrases  with original product names improves classification accuracy by almost 50%.
- To conclude, use of user generated content for classification, though causes ambiguous classification, has a positive influence on grouping products in a more user friendly way.

# Dataset Description

The dataset for the product classification problem consists of 100 product categories, containing 1000 items corresponding to each category. Each product in the dataset consists of the following features:

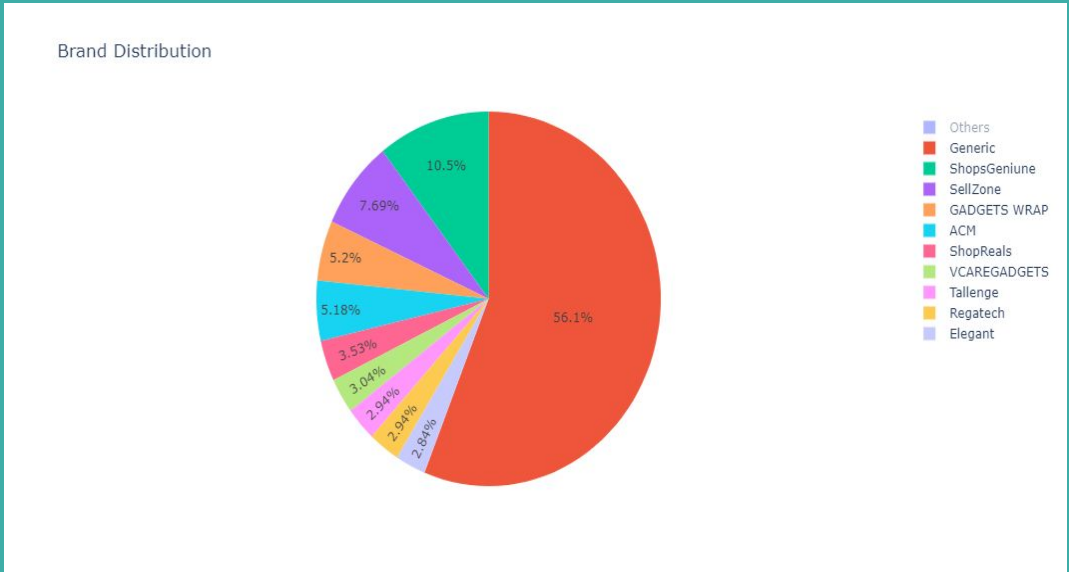
1. Title (String): Title of the product
2. Description (String)
3. Brand Name (String)
4. Bullet points (List): Important features regarding the product

The bullet points for each data item that displays product descriptions in a more comprehensive manner. These columns are evaluated together to make accurate product category predictions.

# Data Pre-processing

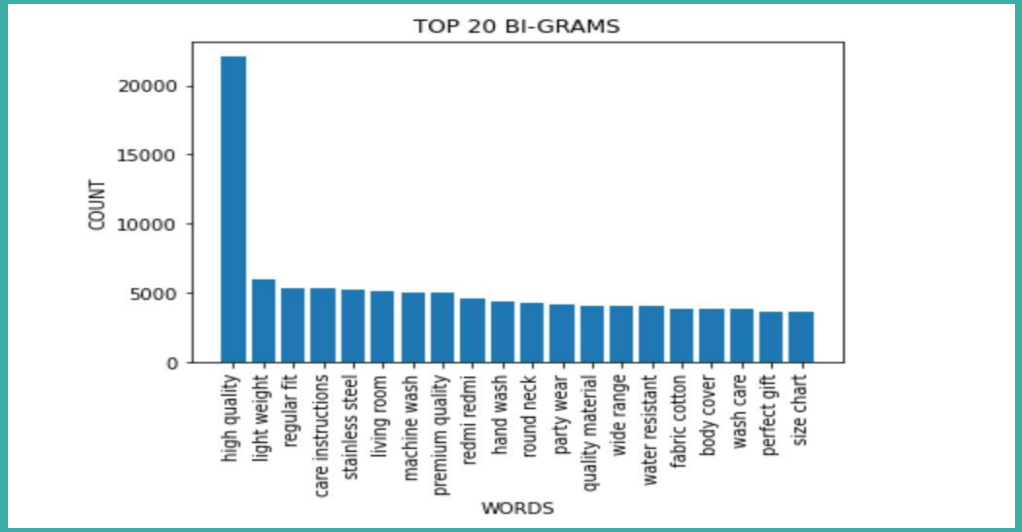
- Null valued columns were eliminated from the data.
- Each data point was converted into an independent document, a tuple containing the product title, description and bullet points together. Further evaluation was performed on the obtained list of tuples.
- Each of these tuples was split further by category, i.e. tokenization was performed on the text document. Tokenization enabled us to perform POS tagging and lemmatization on the dataset based on the category.
- Every document tuple was cleaned, and stop words and punctuations were eliminated. Additionally, extra spaces in the product descriptions were removed. After performing initial processing, each word in the tokenized document was processed as follows. In case the word was a stop word or alphanumeric, it was eliminated from the tuple.
- 'Generic' category products were eliminated. Additionally, it was observed from the model evaluation that words of length less than two do not add any value to the description. Hence, all such instances were eliminated.
- Lastly, words that appear in about 95% of the data points were eliminated from the dataset to improve model performance. This is because such words do not help us differentiate between categories or add any overall value to model predictions. This concluded the comprehensive data preprocessing.

# Data Visualization



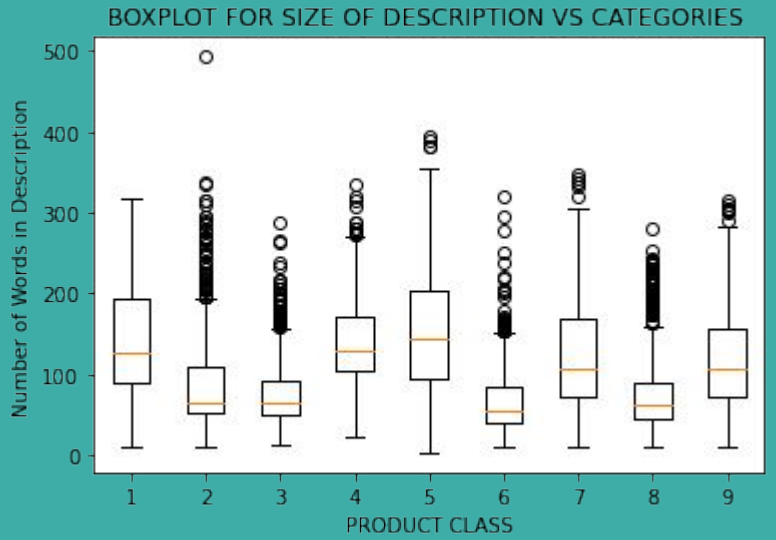

1.1 Brand VS Number of Products


1.2 Word Cloud displaying words with highest occurrence across categories.




1.3 Top 20 bigrams across categories

1.4 Size of description across categories



# Methodology

The count vectorizer was used to convert text data into its appropriate vector representation based on its frequency. On the obtained vectors, we trained suitable classification models.

1. **Logistic Regression:** The parameters that have been adjusted are **penalty, max_iter, C, and solver**. In SGD the tweaked parameters are **alpha, loss, average, eta0, learning_rate, max_iter**.

Logistic Regression (Loss:None)

a. penalty : “none
b. max_iter : 500

Logistic Regression (Loss: L1)

a. penalty : “l1”
b. Solver = “liblinear”
c. C = 0.1
d. max_iter : 500

Logistic Regression (Loss: L2)

a. penalty : “l2”
b. C = 0.01
c. max_iter : 500

SGD Classifier

a. alpha : 0.001
b. average = True
c. loss = log
d. eta = 0.01
e. Learning_rate = “constant”
f. max_iter : 500

2. **Naive Bayes:** The parameters that have been adjusted is **alpha** in Multinomial Naive Bayes and **alpha, binarize** in Bernoulli naive bayes.

Multinomial NB:

alpha : 5

Bernoulli NB:

alpha : 0.01, binarize = 0.5

# Random Forests And Decision Tree

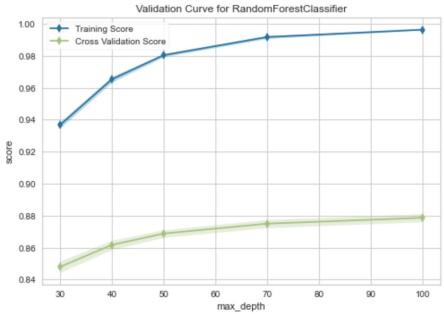
The appropriate values of hyperparameters are chosen after visualisation with validation_curves. The values chosen are:

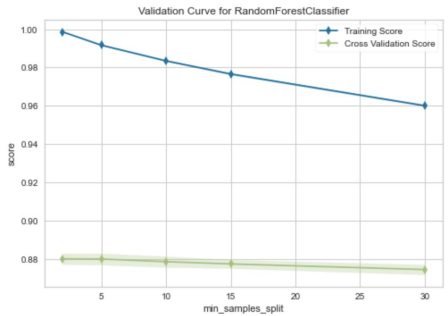
RANDOM FORESTS

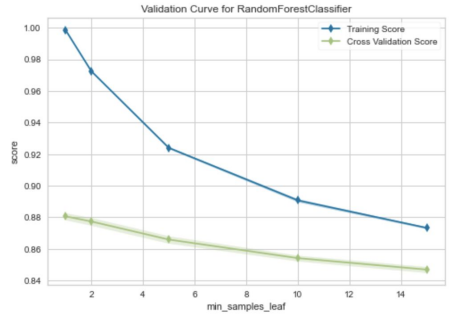
a. n_estimators : 200
b. max_depth : 100
c. min_samples_split : 5
d. min_samples_leaf : 2

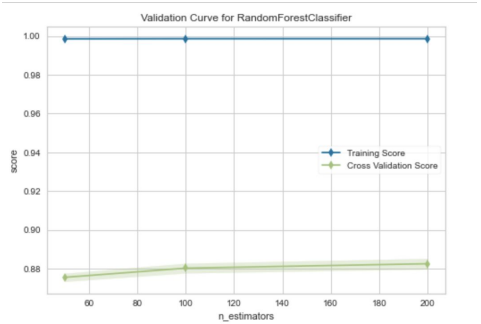
DECISION TREES

a. max_depth : 150
b. Min_samples_split : 5
c. Min_sample_leaf : 1



# SVM and Neural Network

The appropriate values of hyperparameters are chosen after visualisation with Grid Search CV. The values chosen are:

Neural Networks

a. hidden_layer_sizes=[64, 32]
b. activation='logistic'
c. max_iter=100

SVM

a. C = 1
b. Tol = 0.001

KNN

a. n_neighbors = 8

# Results and Analysis

| | Training Data | | | | Testing Data | | | |
|---|---|---|---|---|---|---|---|---|
| **Model** | A | P | R | F1 | A | P | R | F1 |
| **Logistic Regression (Loss: L2)** | 0.91 | 0.91 | 0.90 | 0.90 | 0.87 | 0.87 | 0.87 | 0.87 |
| **Logistic Regression (Loss: L1)** | 0.89 | 0.89 | 0.89 | 0.89 | 0.86 | 0.86 | 0.86 | 0.86 |
| **Logistic Regression (Loss: None)** | 1.00 | 1.00 | 1.00 | 1.00 | 0.84 | 0.84 | 0.84 | 0.84 |
| **SGD Classifier** | 0.89 | 0.89 | 0.89 | 0.89 | 0.86 | 0.86 | 0.86 | 0.86 |
| **Multinomial Naïve Bayes** | 0.82 | 0.83 | 0.82 | 0.82 | 0.80 | 0.80 | 0.80 | 0.79 |
| **Bernoulli Naïve Bayes** | 0.79 | 0.81 | 0.79 | 0.80 | 0.75 | 0.78 | 0.75 | 0.76 |
| **Decision Tree** | 0.95 | 0.97 | 0.97 | 0.97 | 0.81 | 0.82 | 0.81 | 0.82 |
| **Random Forest** | 0.96 | 0.97 | 0.97 | 0.97 | 0.88 | 0.88 | 0.88 | 0.88 |
| **MLP** | 1.00 | 1.00 | 1.00 | 1.00 | 0.86 | 0.86 | 0.86 | 0.86 |
| **K-NN** | 0.86 | 0.85 | 0.85 | 0.85 | 0.79 | 0.76 | 0.76 | 0.77 |
| **SVM** | 0.92 | 0.92 | 0.92 | 0.92 | 0.85 | 0.85 | 0.84 | 0.84 |

# Learnings

- Through this project, we applied various machine learning models, which helped us obtain a clear understanding of concepts.
- We discovered how different models perform better depending on the type of data and tackled various challenges like over-fitting and under-fitting of models.
- We utilized word vectorizers and lemmatizers which helped us understand the optimal procedure towards applying NLP.
- We learnt that product descriptions provide the maximum information alongside the brand name in the product classification task.
- The metric scores are highly promising and show a deep resemblance to the previous work done in the field.
- We were able to provide a viable solution to the product classification task in terms of a model that achieved high accuracy alongside decent recall and precision scores.

# INDIVIDUAL MEMBERS CONTRIBUTION

## Harshita Srinivas

- Data Extraction and Cleaning
- Decision Trees
- Random Forests
- Neural Networks

## Saksham Mehla

- Data Extraction and Cleaning
- Logistic Regression
- SGD Classifier
- Naive Bayes
- KNN

## Hardik Dudeja

- Data Preprocessing and Visualization (Feature Extraction using NLP)
- Decision Trees
- Random Forests
- Neural Networks

## Mohit Bhar

- Data Preprocessing and Visualization (Feature extraction using NLP)
- Logistic Regression
- SGD Classifier
- Naive Bayes
- SVM

**Every member has contributed equally to the project till now.**