Product classification using Textual data

Chaitanya Ashish Khot
Student Number – 22262858
School of Computing
Dublin City University
chaitanya.khot2@mail.dcu.ie

***Abstract*— This study aimed to develop a machine learning model for categorizing products using three different algorithms: Multinomial Naive Bayes, Random Forest, and Logistic Regression. The study includes categorizing the products into various categories using a dataset provided by Etsy with information including titles, descriptions, and tags. This study's main goal was to evaluate the three algorithms' performance in F1 scores and use the top method on the test data. The study started by preprocessing the dataset, which included removing duplicates, null values, and stop words and tokenizing the words. The performance of this algorithm was compared to that of Random Forest and Logistic Regression, with Multinomial Naive Bayes being used as the baseline model. F1 scores were one of the evaluation criteria utilized in this study to compare the effectiveness of the algorithms.**

Keywords—product classification, multinomial naïve bayes, random forest, logistic regression, batching

I. INTRODUCTION

Product classification is a vital task in e-commerce since it helps companies to give their customers relevant search results and recommendations. However, manually classifying many products can be time-consuming and error-prone. Automation of the classification process using machine learning algorithms produces more accurate and effective outcomes. With the help of three different algorithms—Multinomial Naive Bayes, Random Forest, and Logistic Regression—we hope to create a machine learning model for categorizing products using Etsy's product data in this study. The dataset used for the study has attributes like title, description, and tags to categorize products into several groups. To give a thorough knowledge of the performance of the algorithms, we use F1 scores as our primary evaluation metric.

At first, Multinomial Naive Bayes is used as the baseline model, and compare its performance with that of Random Forest and Logistic Regression. Our findings highlight the efficacy of each algorithm for product classification and by using F1 ratings to assess each algorithm's performance appropriately.

Future research can use the study's findings to enhance their classification procedures and give their clients better search results and recommendations. As machine learning algorithms continue to be used more and more frequently in the e-commerce sector to automate classification jobs, it also offers a roadmap for future study in this area.

II. DATASET

We received textual and image data from Etsy as TF Records and Parquet files. The information utilized for this study is in parquet format. It contains several attributes for products, like their product id, title, description, tags, and so forth, to categorize them into distinct groups. It is a structured dataset with 245,485 rows and 21 characteristics. The data is pre-processed to remove any invalid or missing values before being used to train the model. The number of non-null values and their data types are displayed in the table below before the data is pre-processed.

#	*Column*	*Non-Null Count*	*Data type*
0	product_id	245485	int64
1	title	244545	object
2	description	244545	object
3	tags	210575	object
4	type	244211	object
5	room	8727	object
6	craft_type	32520	object
7	recipient	13753	object
8	material	20876	object
9	occasion	53229	object
10	holiday	41019	object
11	art_subject	2773	object
12	style	17032	object
13	shape	2358	object
14	pattern	10678	object
15	bottom_category_id	245485	int64
16	bottom_category_text	245485	object
17	top_category_id	245485	int64
18	top_category_text	245485	object
19	color_id	245485	int64
20	color_text	245485	object

Table. 1. Non-null value counts in the train dataset

III. RETATED WORK

The research [1] uses machine learning approaches to present a novel method for extracting and classifying product attributes and text emotion analysis. The suggested approach involves feature extraction from the data, model construction, and data pre-processing. The accuracy, F1 score, precision, and recall were some of the evaluation measures the authors used to assess the performance of the suggested strategy. The study's findings are encouraging, as the proposed strategy outperforms current ones regarding the accuracy and F1 score. This research offers valuable insights into applying machine learning approaches for text emotion analysis, product attribute extraction, and classification.

In this study [2], the performance of the Multinomial Naive Bayes and Bernoulli Naive Bayes text classification Naive Bayes algorithms is compared. The performance of the two algorithms is compared in the study using data

preprocessing, feature extraction, and model development. The authors assessed the algorithms' performance using criteria for precision, recall, F1 score, and accuracy. According to the study, the Multinomial Naive Bayes algorithm performs better than the Bernoulli Naive Bayes algorithm for text categorization regarding F1 score and accuracy. The research's conclusions give crucial new information about how Naive Bayes algorithms for text classification can be used to classify product categories using machine learning methods.

In the work done by [3], the performance of two well-known machine learning algorithms, Support Vector Machine (SVM) and Naive Bayes, for sentiment analysis of product reviews. The study's approach comprises feature extraction from the data, model creation, precision, recall, F1 score, and accuracy measures for evaluation. According to the study, the accuracy and F1 scores of the SVM and Naive Bayes algorithms are comparable. This study offers insightful information on using sentiment analysis to machine learning algorithms, which may be used to classify products based on text input.

A method for classifying products based on text data using logistic regression and part-of-speech tagging is suggested in the research [4]. The project uses a dataset of product reviews to classify the evaluations into various groups using logistic regression and part-of-speech tagging. The performance of the proposed methodology is compared in the study to that of other machine learning algorithms like Naive Bayes and Support Vector Machine. Accuracy, precision, recall, and the F1-score are the evaluation criteria employed. The findings demonstrate that, in terms of accuracy and F1 score, the suggested methodology performs better than the other algorithms. The study concludes that using part-of-speech tagging with logistic regression can improve the accuracy of text classification, which is relevant for product category classification on text data using machine learning.

The work done by Kanish Shah et al. In [5] gives a comparative analysis of three well-known machine learning models for text classification—Logistic Regression, Random Forest, and K-Nearest Neighbor (KNN) The authors assessed these models' performance using four distinct datasets and various evaluation criteria, including precision, recall, F1-score, and accuracy. They have also undertaken feature selection to determine the most crucial features for classification. Regarding accuracy and F1-score, the results show that Random Forest outperforms the other two models. The authors advise using Random Forest for text classification tasks, especially when working with unbalanced datasets.

IV. Methodology

A. Initial data exploration and data cleaning

We identified that the provided data had over 70 percent missing values in some columns like material, occasion, and holiday as shown in the graph below.

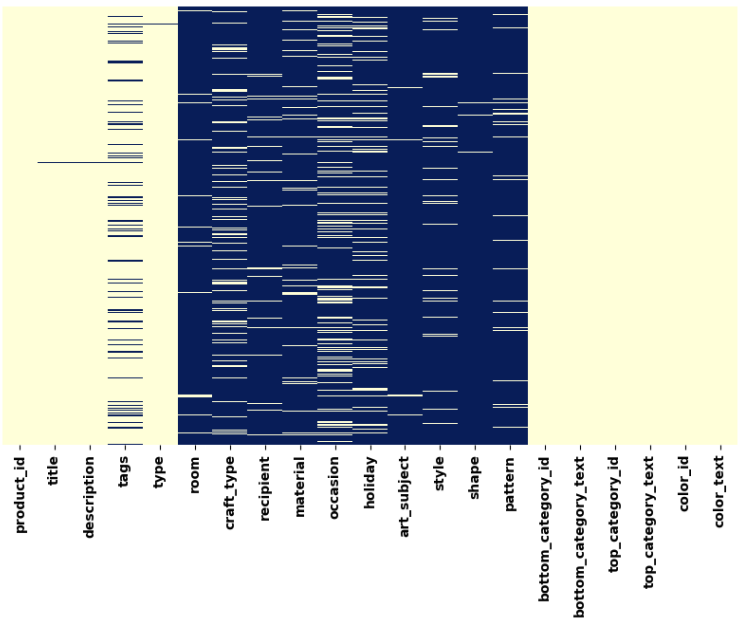

Fig. 1. Missing data matrix for all the features in the dataset

- Handling columns: It was discovered that attempting to impute values into columns with a large percentage of missing data could result in inaccurate data and hinder the model's training. As a result, columns with more than 70% of their data missing were removed.
- Handling NaN values: After doing a NaN value check on the remaining columns, it was discovered that the tags column had almost 14% of its data missing, compared to less than 1% for the title, description, and type columns. In theory, values in the tags column should be imputed, but we had no method of confirming whether the imputed values were valid, which would have negatively impacted the outcomes of our model. Additionally, there were more than 200,000 rows of non-null data. Hence, all the null values were dropped from the data.

After handling these issues, the clean data consisted of tags, title, description and the type columns along with all the target variables.

B. Data preprocessing and feature engineering

These cleaned columns had texts like URLs and emoji, which required pre-processing. Hence, we defined functions to pre-process the texts using Natural language processing as follows:

- First, all of the text was made lowercase. We also eliminated numbers, URLs, special characters, emojis, and other characters from the text.
- Next, the text was cleaned up by removing terms like "a" and "the." Additionally, we eliminated instances where new lines appeared in the form of '\n'.
- The text was then tokenized, and this tokenized text was then subjected to a word lemmatizer. This step assisted in getting rid of all the additional spaces in the text and returned it to its original shape.

These were then recorded as new columns with the prefix "processed" for the appropriate columns and applied as

necessary to the "title," "tags," and "description" columns. This processed data was saved as a backup in case the processed data was directly needed.

V. MODEL SELECTION

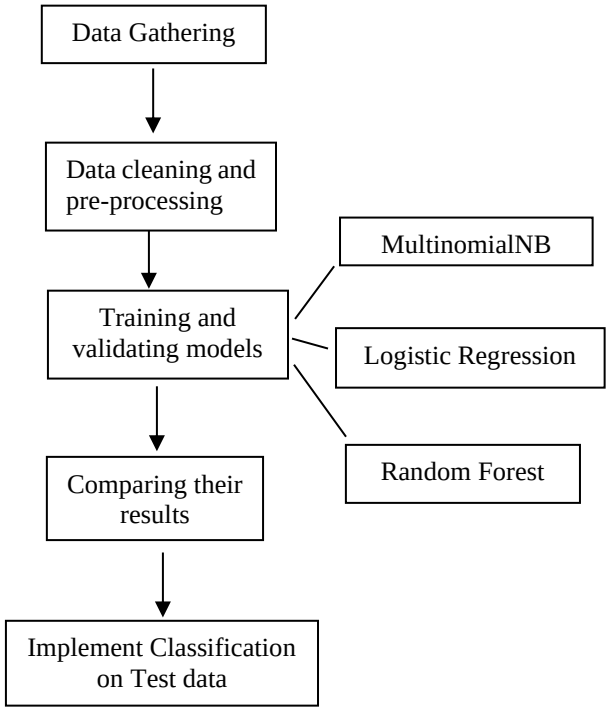

Fig. 2. Flowchat of the project

Multinomial Naive Bayes (MNB), which is straightforward and efficient for text classification tasks, served as our baseline model. Because MNB is frequently used for text classification tasks and has a good track record on sizable datasets, we selected it as our baseline model. However, it might not be the optimal model because our dataset includes numerous linked features (such as title and tags) and because MNB requires that features are independent and identically distributed. As a result, we also tested several alternative models, including Logistic Regression and Random Forest.

Random Forest is an ensemble learning algorithm that generates many decision trees and produces the class representing the mean prediction made by each tree or the mode of the classes. It is a good option for our product classification task because of its reputation for handling noise. However, despite the model being tuned for various hyper parameters, there appeared to be a significant data gap between the test and validation data sets.

Last but not least, a statistical model known as logistic regression is used to examine the relationship between a dependent variable and one or more independent variables. It is well-regarded for being straightforward and comprehensible and is frequently used in classification assignments. On both the train and validation sets, it was discovered that logistic regression provided the best results for this data. For added assurance, we visualized a learning curve.

Overall, we experimented with various machine learning algorithms to build a product classification model that performs well on our dataset. We started with a simple baseline model (MNB) and then explored more complex models, such as Random Forest and Logistic Regression. By comparing the performance of these models based on their F1 scores, we identified that Logistic regression is the best algorithm for our classification problem.

A. Experiments and parameter tuning:

Due to the data size—more than 200,000 records—Google Colab's processing capacity could not simultaneously process and train the entire data set. As a result, all of the data was trained in batches, which helped minimize the needed processing power and shorten the training period. Our batched data were processed in parallel using the Swifter package as well. The data was trained using a variety of batch sizes, models, and training parameters, including the processed title, tags, and description, as well as their combinations. It was also discovered that the most significant results came from solely using the processed "title" column.

In order to scale the data for **Multinomial Naive Bayes**, the data was first vectorized using TF IDF and then scaled using MinMaxScaler, which scales data between 0 and 1. Since MNB only accepts non-negative data while StandardScaler scales the data from -3 to +3, MinMaxScaler was used instead of StandardScaler.

For this model, we used different input parameters and their combinations and batch sizes of 10,000, 15,000, and 20,000. In order to arrive at an overall f1 score, we finally concatenated all the batch data. With a batch size of 20,000, we discovered that MNB delivered the best results (see the results table below). Although there was a data gap between the train and validation sets, the findings were still satisfactory.

The model tends to overfit the training data as batch size increases, according to our experimentation with **Random Forests** with batches of 10,000, 15,000, and 20,000. This might be because random forests need a clear idea of batching and need to be optimized with it. In order to achieve an appropriate fit, we additionally tested various n_estimaters and max depth. Using 200 trees (n_estimators) and 30 max_depth, random forests produced the best results on a batch of 15,000 on the "title" column (see the results table below). The outcomes could have been better even if we decreased overfitting.

We also tried out **Logistic regression** at the end. Because logistic regression supports non-negative data, we used TF IDF vectorizer and MinMaxScaler before applying the model, just like MNB. In addition, we used a chi-squared test to apply a feature selector that uses the most prominent features. With a regularization strength of 0.2 and a cap of 1000 iterations, we also used L2 regularization. A batch size of 5,000, 10,000, 20,000, and 30,000 was used for this technique. Following model implementation, we displayed a learning curve for various batch sizes and their validation and f1 scores. Thanks to this visualization, we realized that 10,000 is the ideal batch size for logistic regression. Finally, we implemented this model with a batch size of 10,000 logistic regression, providing the best and most consistent results on train and validation sets. Below figures (Fig.3, Fig. 4, Fig. 5) show the learning curves for training and validation F1 scores of Bottom Category ID, Top Category ID and Color ID respectively.

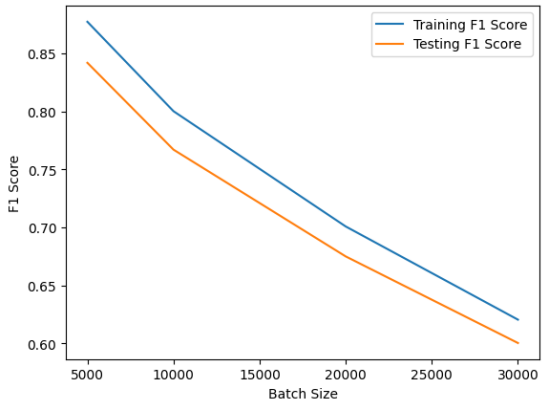

Fig. 3. Learning curve for bottom category ID

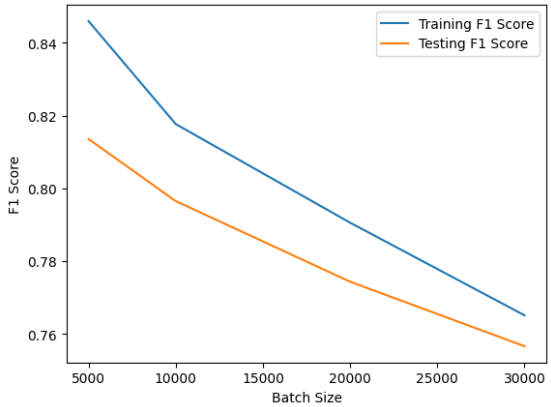

Fig. 4. Learning curve for top category ID

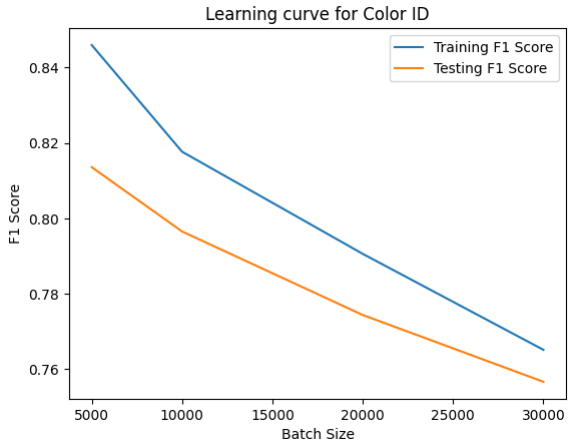

Fig. 5. Learning curve for color ID

B. *Evaluation and results:*

Since F1 scores are a more accurate evaluation measurement than accuracy, we used them to evaluate all of these models. The table below (Table. 2.) includes all of the outcomes. Although all models produce comparable results, logistic regression consistently delivers strong outcomes with the least amount of data gaps and no overfitting. So, to forecast our top product id, bottom product id, and color id for our final test data, we apply logistic regression.

Model Name	**Data Type**	**Bottom Category ID F1 score**	**Top Category ID F1 score**	**Color ID F1 score**
Multinomial NB	Train	96.76%	83.93%	56.76%
	Validation	72.47%	69.87%	26.56%
Random Forest	Train	88.94%	74.19%	60.42%
	Validation	71.28%	67.30%	45.26%
Logistic Regression	Train	80.02%	81.76%	46.00%
	Validation	76.70%	79.65%	43.49%

Table. 2. Results table showing the F1 Scores for training and validation datasets

C. *Limitations and Future work*

This study only uses textual data stored in parquet files. Additionally, we can observe that our model has trouble correctly predicting the color id. This prediction occurs because our textual columns does not always contain the color mentioned. We can over-sample the data and retrain our model for better color identification results.

Convolutional Neural Networks (CNN) are one example of an image classification approach that may be used with the provided image data to get improved results. However, TF Records are large files that take a lot of processing power to read and analyze.

VI. Conclusion

In summary, by using a dataset provided by Etsy, this study successfully implemented three machine learning algorithms—Multinomial Naive Bayes, Random Forest, and Logistic Regression—and compared their performance in classifying products into top product id, bottom product id, and color id categories. In terms of consistency and accuracy, the results showed that Logistic Regression performed better than the other two methods while being less prone to data gaps or overfitting. The performance of Logistic Regression on the provided dataset for several batches was also evaluated by creating a learning curve for it. The study offers beneficial insights for companies in the e-commerce sector to enhance their product classification method and offer customers better search results and recommendations.

References

[1] C. Wen, J. Wu, and D. Chen, "Analysis of Text Emotion Based on Logistic Regression Model," *IEEE Xplore*, Nov. 01, 2022. https://ieeexplore.ieee.org/document/9994346/ (accessed Apr. 17, 2023).

[2] G. Singh, B. Kumar, L. Gaur, and A. Tyagi, "Comparison between Multinomial and Bernoulli Naïve Bayes for Text Classification," *IEEE Xplore*, Apr. 01, 2019. https://ieeexplore.ieee.org/document/8776800/

[3] S. Rana and A. Singh, "Comparative analysis of sentiment orientation using SVM and Naive Bayes techniques," *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, Oct. 2016, doi: https://doi.org/10.1109/ngct.2016.7877399.

[4] T. Pranckevicius and V. Marcinkevicius, "Application of Logistic Regression with part-of-the-speech tagging for multi-class text classification," *2016 IEEE 4th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, Nov. 2016, doi: https://doi.org/10.1109/aieee.2016.7821805.

[5] K. Shah, H. Patel, D. Sanghvi, and M. Shah, "A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification," *Augmented Human Research*, vol. 5, no. 1, Mar. 2020, doi: https://doi.org/10.1007/s41133-020-00032-0.