Classification And Prediction Of Product Attributes for Etsy Marketplace Using Machine Learning Algorithms

Raj Vibhute
(22263276)
School of Computing
Dublin City University, Ireland
raj.vibhute2@mail.dcu.ie

1. ABSTRACT:

The aim of this project is to learn and predict patterns for the Etsy marketplace. The end goal was to train an ML model to predict top category id, bottom category id and colour id. The model trained and implemented is the multinomial Naïve Bayes algorithm. The model was tested and trained using a multi-output classification method. The F1 metric is used to evaluate the model. The F1 score for the top_category_id is 0.544, the F1 score for the bottom_category_id is 0.526, and the F1 score for the color_id is 0.262.

2. INTRODUCTION:

Online shopping is becoming more and more popular among consumers in the current digital era thanks to the rapid development of e-commerce. Because of this, e-commerce has emerged as a key channel for companies to connect with consumers. Predicting product qualities is crucial for an e-commerce business to increase search accuracy and consumer happiness. Identifying significant traits and qualities of a product based on a variety of data sources, such as the title, product description, and product material [2].

Machine learning algorithms have gained popularity as a method for predicting product qualities in the e-commerce sector. These algorithms have the capacity to analyze vast volumes of data and discover links and patterns that may be applied to precisely anticipate product qualities. Natural language processing techniques can be used in conjunction with machine learning approaches to extract meaningful information from textual input. [2].

In this study, we use machine learning and natural language processing methods to create a product attribute prediction model for Etsy. In order to forecast important product properties, our method includes training a model using a sizable collection of product data, including some features of the product. The major objective is to develop a model that accurately predicts product characteristics, which enhances search precision and makes it possible for clients to identify the products that satisfy their demands.

3. RELATED WORK:

In their innovative task of discriminative attribute extraction, V. Embar et al. [1] suggest determining the characteristics that separate product variations and extracting the values of those characteristics from unstructured text. The article offers a deep learning-based method called DiffXtract that uses a multitask objective and explicit semantic representation to simultaneously identify the qualities and extract their values.

L. Sun et al. [2] provide a machine learning-based method for extracting and categorizing product attribute phrases from internet reviews. The method identifies out-of-vocabulary attribute terms and corrects word segmentation findings using a word-internal tag mechanism. The strategy uses the word-level text classification method based on distributed word representation to categorize the product attribute terms using the word-internal tag technique.

A method for creating a fashion style classification system utilizing supervised learning techniques is put forth by C. David Kreyenhagen et al. [3] in order to enhance customer search functionality and offer tailored recommendations in online fashion shopping. To assign numerical values to each brand in the training set of brand style associations, they are utilizing natural language processing and text mining. After that, a support vector machine is trained on this training set to classify styles.

In their study, Wirojwatanakul, P. et al. [4] suggest examining multi-modal methods for classifying e-commerce items on Amazon utilizing titles, descriptions, and photos. The strategy intends to create a tri-modal late fusion model with superior single-modal models at product classification. The study demonstrates that the inadequacies of each modality can be supplemented by integrating various modalities, resulting in enhanced performance in multi-label classification problems.

Convolutional neural networks (CNNs) are the focus of S. Dara et al.'s paper [5], which aims to give an overview of the feature extraction techniques used in deep learning. The success of deep learning applications is emphasized by the study as being highly dependent on feature extraction. The authors want to provide readers with a thorough knowledge of where feature extraction is at right now in deep learning and what it means for the direction of subsequent studies in the area.

Using deep learning models that have been previously trained on substantial corpora, la Comble, A. D. et al. [6] suggest a technique for forecasting product features from unstructured data. They outline their process for extracting attributes, which includes looking into single-modality techniques and creating a multi-modal model. They suggest a cutting-edge method for seamlessly fusing modalities.

For multinomial text classification, N. Sharma et al. [7] present a modified version of the naive bayes classifier. The altered model seeks to overcome the shortcomings of the traditional bag of words concept. To enhance the performance of the Naive Bayes classifier, a combination of feature selection and term weighting is recommended.

4. EXPLORATORY DATA ANALYSIS

It is crucial to first study the data and create intuitions utilizing the analysis before proceeding to the section of the prediction that involves real outcomes. 'Train data' and 'test data' are the two main categories of data. 245 485 rows make up the training data, while 27 119 rows make up the testing data. We can train our model on a relatively big sample size in this situation, which may be beneficial. For each product, the training data includes the following attributes: -

- product_id
- title
- description
- tags
- type
- room
- craft_type
- recipient
- material
- occasion
- art_subject
- style
- shape
- pattern
- bottom_category_id
- bottom_category_text
- top_category_id
- top_category_text
- color_id
- color_text

Except for bottom_category_id, top_category_id, and color_id, which we must predict, the test data shares all other attributes with the train data. Some of the features for the training dataset must be pre-processed before the prediction is made since the values inside these features contain several extraneous characters that will impair the prediction. Additionally, there are some missing values in the data that must be addressed. There are three distinct entries under "type," which are physical, None, and download. Twenty distinct entries total in color_id. We calculated the percentages of missing data for each feature and discovered that attributes like room, recipient, material, art_subject, style, shape, and pattern had more than 90% of their values missing, whereas the craft_type attribute had about 86%, the occasion had about 78%, the holiday had about 83%, and tags had about 14%. A very little quantity of data is also missing from features like title, description, and kind.

5. DATA PRE-PROCESSING

5.1 REMOVING COLUMNS:

The shape, room, recipient, holiday, and art_subject attributes were among those we removed from the dataset after calculating the percentage of data that was missing for each attribute. These attributes had over 95% missing values, while the recipient wouldn't be very helpful in making predictions and had around 94% missing values. The same was true for the holiday attribute, which won't have an impact on our predictions and had 83% missing values.

5.2 FILLING NULL VALUES:

We have now filled in the null values from the title, description, tags, and type properties after eliminating a few attributes from our dataset. These characteristics contain a very small number of null values; thus, we have replaced them with empty strings. We have substituted the mode for the null values for the attributes craft_type, material, occasion, style, and pattern.

6. METHODOLOGIES

6.1 NAÏVE-BAYES CLASSIFIER:

The Bayes theorem is used by the well-known classification technique known as Multinomial Naive-Bayes to categorise the text. It is an algorithm for supervised learning. To learn from it, it needs labelled training data. It offers the capability of classifying data that cannot be mathematically represented [11].

The fundamental idea behind this technique is to describe each document as a collection of words where the word order is irrelevant. A feature is the frequency of each word. Using the Bayes theorem, the algorithm then determines the likelihood of each class given the document's feature values [8].

6.2 COUNT-VECTORIZER:

Characters and words are not understood by machines. So, for a machine to understand text data, it must be represented in numerical form. Here, CountVecorizer is used. It is a tool made available by the Python scikit-learn module. It is a well-known text-processing method for turning a list of words into a matrix of token counts [12].

The goal of CountVecorizer is to extract a vocabulary of original words from the text documents, after which each document is represented as a vector of word frequencies. The final matrix can then be fed into machine learning algorithms for tasks like text classification and other NLP tasks [12].

6.3 TF-IDF TRANSFORMER:

Term-frequency times inverse document-frequency which is also referred as TF-IDF, is a typical term weighting technique used in information retrieval that is also effective in classifying documents. To reduce the influence of tokens that appear often in a corpus and are thus experimentally less useful than features that occur in a tiny portion of the training corpus, TF-IDF is used instead of raw frequency of occurrence of a token in a specific document [13].

7. MODEL TRAINING:

Here, we've trained a multi-output classification model to forecast the top_category_id, bottom_category_id, and color_id target variables for a dataset of product listings. First, we loaded various modules from Scikit-Learn, such as MultimonialNB, CountVecorizer, TfidfTransformer, and train_test_split. These modules will be used to create a multi-output classification model, divide the dataset into training and testing sets, and assess the model's effectiveness.

After importing the above-mentioned Python modules, we used the train_test_split function to divide the data frame 'df_train' into training and testing sets. A random state of 42 is also utilised to ensure reproducibility. The data is divided into an 80:20 ratio, where 80% of the data from the data frame "df_train" will be used to train our model and 20% of the data will be used for testing. 196,388 products will be utilised for training, while 49,097 products will be used for testing.

Following that, the model's features were established as a combination of nine separate fields: title, description, tags, type, craft_type, material, style, pattern, and occasion. Using the apply method from pandas, these features were combined and converted into a single text input.

Then the model was subsequently trained using a pipeline consisting of the three steps CountVectoizer, TfidfTransformer, and MultiOutputClassifier. The CountVectorizer aids in the conversion of the raw text data into a word count matrix where each row in the matrix represents the product and a column represents a distinct word in the dataset. Following that, the word count matrix is transformed into TF-IDF values via the TfidfTransformer. A Naive-Bayes classifier is wrapped in a multi-output wrapper by the MultiOutputClassifier once the pipeline has been trained on the training features, enabling the model to predict all three target variables simultaneously.

8. EVALUATION:

We used the F1-score, a widely used metric for assessing classification models, to assess the effectiveness of our model. Additionally, for each label, we have prepared a classification report. Top_category_id, Bottom_category_id, and Color_id are the three categories for which we computed the F1-score. The macro-averaged F1 score for our model has also been determined.

The F1 score for the three attributes are as follows :

- top_category_id: 0.544
- bottom_category_id: 0.526
- color_id: 0.262.

Additionally, the macro-averaged F1-score, which gives each label equal weight, is 0.444.

9. CONCLUSION:

To enable e-commerce businesses better understand client preferences, we have proposed in this study a task of product attribute classification. Data pre-processing on the given dataset was an essential part of this project for training the model and performing relevant predictions for the chosen attributes. Multinomial Naive-Bayes is the model currently in use for text classification. The model was found to have moderate F1 scores for each target variable, indicating that it can categorize the products according to their properties. In product recommendation systems, our study offers a potential method for forecasting the target variables.

One fundamental method we employed in our research was multinomial Nave-Bayes. Other, better classification techniques, such as Support Vector Machine, Random Forest, Gradient Boosting, and Recurrent Neural Networks, can be used in future studies. Given that Multinomial Naive-Bayes is a far more simplistic model than these, there is a probability that using them will yield superior results. These are the inferences that can be made from the product attribute categorization and prediction project that has been put in place.

10.REFERENCES

[1] V. Embar, A. Kan, B. Sisman, C. Faloutsos and L. Getoor, "DiffXtract: Joint Discriminative Product Attribute-Value Extraction," 2021 IEEE International Conference on Big Knowledge (ICBK), Auckland, New Zealand, 2021, pp. 271-280, doi: 10.1109/ICKG52313.2021.00044.

[2] L. Sun, "Research on Product Attribute Extraction and Classification Method for Online Review," 2017 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), Wuhan, China, 2017, pp. 117-121, doi: 10.1109/ICIICII.2017.37.

[3] C. David Kreyenhagen, T. I. Aleshin, J. E. Bouchard, A. M. I. Wise and R. K. Zalegowski, "Using supervised learning to classify clothing brand styles," 2014 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA, 2014, pp. 239-243, doi: 10.1109/SIEDS.2014.6829909.

[4] Wirojwatanakul, P., & Wangperawong, A. (2019, June 30). *Multi-Label Product Categorization Using Multi-Modal Fusion Models*. arXiv.org. https://arxiv.org/abs/1907.00420v2.

[5] S. Dara and P. Tumma, "Feature Extraction By Using Deep Learning: A Survey," 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2018, pp. 1795-1801, doi: 10.1109/ICECA.2018.8474912.

[6] la Comble, A. D., Dutt, A., Montalvo, P., & Salah, A. (2022, March 7). *Multi-Modal Attribute Extraction for E-Commerce*. arXiv.org. https://arxiv.org/abs/2203.03441v.

[7] N. Sharma and M. Singh, "Modifying Naive Bayes classifier for multinomial text classification," 2016 International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, India, 2016, pp. 1-7, doi: 10.1109/ICRAIE.2016.7939519.

[8] Kibriya, A. M., Frank, E., Pfahringer, B., & Holmes, G. (n.d.). *Multinomial Naive Bayes for Text Categorization Revisited*. Multinomial Naive Bayes for Text Categorization Revisited | SpringerLink. https://doi.org/10.1007/978-3-540-30549-1_43

[9] Su, J., Sayyad-Shirabd, J. and MAtwin, S. (2011). Large Scale text classification using semi-supervised multinomial naive bayes. *ICML'11: Proceedings of the 28th International Conference on International Conference on Machine Learning*, [online] pp.97–104. Available at: https://dl.acm.org/doi/abs/10.5555/3104482.3104495.

[10] Xu, S., Li, Y., & Wang, Z. (2017). Bayesian Multinomial Naïve Bayes Classifier to Text Classification. *Lecture Notes in Electrical Engineering*, 347–352. https://doi.org/10.1007/978-981-10-5041-1_57.

[11] Ratz, A. V. (2022, April 8). *Multinomial Naïve Bayes' For Documents Classification and Natural Language Processing (NLP)*. Medium. https://towardsdatascience.com/multinomial-na%C3%AFve-bayes-for-documents-classification-and-natural-language-processing-nlp-e08cc848ce6

[12] GeeksforGeeks. (2020). *Using CountVectorizer to Extracting Features from Text*. [online] Available at: https://www.geeksforgeeks.org/using-countvectorizer-to-extracting-features-from-text/.

[13] scikit-learn.org. (n.d.). *sklearn.feature_extraction.text.TfidfTransformer — scikit-learn 0.23.1 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html.