Comparitive Analysis of various Machine learning Algorithms for E-commerce Platform

Rohit Shinde
22260213
rohit.shinde2@mail.dcu.ie

Abstract— One of the challenging areas of machine learning is prediction. Over the years, there has been a necessity to practice the machine learning models which are used in different fields other than IT industry such as health, entertainment, education, etc. E-commerce businesses have evolved in recent years and a lot of techniques of machine learnings are implemented in this domain to attract or retain targeted customers. In this paper, we are doing multi class classification for prediction of product categories by performing different machine learning algorithms and attaining maximum accuracy and f1 score.

Keywords— *Text Pre-Processing, Feature Engineering, Machine Learning, Classification, Prediction, F1-score*

1. Introduction

The number of prospective customers and sellers worldwide has significantly increased because of the expansion of business platforms like e-commerce websites. E-commerce has become increasingly popular and in the spotlight thanks to the quick growth of the internet and sophisticated computer technology. Online goods shopping is becoming more popular in daily life, and as a result, website structures are getting more and more sophisticated. The product should be viewable by the user according to their specific preferences, such as product category, description, tags, colour, etc.

Etsy is a huge two-sided online marketplace where individuals can buy, sell, trade, and collect one-of-a-kind products. Predicting the result based on the dataset has been a fascinating topic in machine learning since its inception. Over time, researchers have done numerous studies and experiments to improve the models and algorithms.

In this paper, we have done data pre-processing, feature engineering with different text classification techniques. Then different machine learning algorithms are utilized for multi class classification on train data and tried to increase the f1 score of each attribute and predict some attributes in the given test dataset. The results of all the algorithms are compared to know which one is giving maximum f1 score and finally deciding that model for target attribute.

2. Related work

Fashion businesses must continually adjust to the shifting tastes and wants of their customers because they operate in a fast-changing and intensely competitive environment. In reality, one of the most significant aspects affecting consumers' tastes is the fashion market trends. Paper [1] suggests a method for identifying fashion trends that is based on the approach of experts. There method collects web text data from such sources (using “Beautiful Soup” library) and applies text mining tools to identify the most popular fashion subjects addressed over a specific period of time, as opposed to manually searching through a ton of fashion magazines and weblogs [1]. Natural language processing was used for data preparation and cleaning such as stop words removal, removal of URLs, stemming of words, etc. Paper [2] focuses on text classification problem and to address this problem they have used TF-IDF to evaluate how important a word is to a document in a corpus. For proper fitting on any of the machine learning model the data is normalized using TF-IDF Transformer.

Further, to use the processed data with appropriate machine learning model paper [3] has compared different algorithms such as Logistic Regression, Naive Bayes, Support Vector Machine, Random Forest to attain better accuracy and f1 score. The goal of their framework was to use multiple machine learning techniques to identify diabetes early and save a patient's time and money [3]. This will be implemented in our work to get maximum f1 score for attribute “top_category_id” from our train data and decide which one is best to go with.

For determining the percentage of online products that are defective, innovative logistic regression performs noticeably better and provides more accurate findings than linear regression claims paper [4]. It can give clarity to determine which algorithm to go with for attribute “color_id” in our training dataset.

All the above related work contributes efficiently to get to our main goal of multi class classification and improving the f1 score in given training dataset and predicting the top categories on test dataset.

3. Methodology

3.1 Dataset

The data with training and testing sets are provided by Etsy- a huge e-commerce platform which is used to work on the main problem statement. The data is provided in two formats- parquet and tfrecords. Tfrecords contains image data whereas parquet dataset is textual data. Using parquet dataset to move forward as it is less in the size and requires less computational requirements. Parquet train data consists of 245,485 records and test data has 27,119 records to work on. Train dataset has following columns: product_id, title, description, tags, type, room, craft_type, recipient, material, occasion, holiday, art_subject, style, shape, pattern, bottom_category_id, bottom_category_text, top_category_id, top_category_text, color_id and color_text. For train data, the target features are provided i.e., top_category_id, bottom_category_id and color_id whereas for test data these features are not given in

the test dataset that we need to predict by training our models on train data and then predicting those with test data.

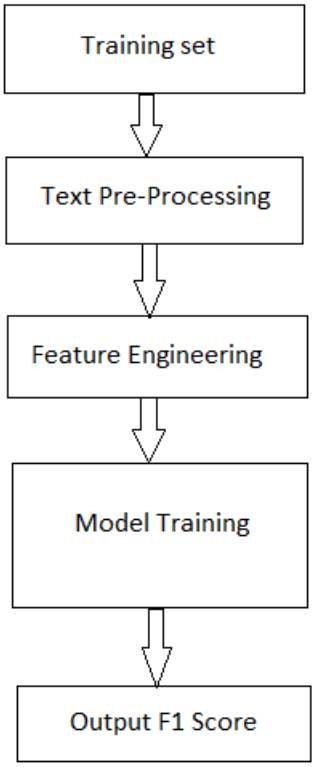

Fig 1. Implementation Flow

3.2 Data Pre-Processing

After analyzing the entire data and found out that the required columns are "title", "description" and "tags". Next step to perform on train dataset is to find out null values and there were 10 columns with more than 80% of null values which was very high hence, dropped those columns as they were not important. "title" and "description" columns had minimal null values so imputed them with "unknown". "tags" column had 14% null values, so it is imputed with keywords from "title" and "description" by using *spaCy* library [5] which is helpful in processing large volumes of text quickly. Column "type" was imputed by analyzing columns "title" and "description", noticed that these columns contain string like "no physical" and "download", if any of the record contains this string in "title" and "description" column then that null record was filled with word "download" in column "type" otherwise it will fill with word "physical" as there were only two unique values present in "type". To prepare high-quality data for processing, it was necessary to undertake various text pre-processing operations. Executed the text pre-processing processes mentioned below on three main columns:

- HTML decoding
- Lowercasing
- Removing stop words and punctuations
- Removing symbols and special characters
- Stemming

3.3 Feature Engineering

Once the data is cleaned then needed to perform feature engineering to get more insights and extracted words for further use of data. Description contained most of the text as it has detailed information of product so extracted all the colors from description which were present in "color_text" column and created new column "extracted_colors" which consists of colors from description. Then again extracted first occurrence of color from description and created new column for that as well named "first_occurrence_color". To get important keywords from column "description" used *nltk* library [5] and extracted 10 keywords which are important and created one more column which consists of these keywords named "keywords".

4. Implementation

Used four models- Naïve Bayes, Logistics regression, linear regression SGD classifier and Random Forest. Created pipeline starting with CountVectorizer which takes a corpus of text documents as input and creates a matrix where each row represents a document, and each column represents a unique word in the corpus. It is used to convert a collection of text documents into a matrix of token counts. Although, performed text pre-processing earlier but on the safer side added it to the pipeline. After this, TfidfTransformer is used as it takes the matrix of token counts generated by CountVectorizer as input and calculates the TF-IDF values for each element in the matrix. The resulting matrix has the same dimensions as the input matrix, but with each element replaced by its corresponding TF-IDF value. This was the common part of the pipeline which is used in all the four algorithms.

Multinomial NB without hyperparameters is executed first with the processed data for top_category_id and color_id separately but it did not gave sufficient f1 scores.

Stochastic Gradient Descent classifier is linear classifier in scikit-learn library which is designed to handle the large dataset and it is used to solve multiclass classification problems. That's why chose this as our problem revolves around multi class classification. Also, it has several hyperparameters such as "loss", "penalty", "alpha" that we tuned to customize its behaviour. For attribute color_id it performed best as compared to other three algorithms as shown in table 2. But for top_category_id it gave adequate results shown in table 1.

Next Logistic Regression is used as it is supervised classification algorithm that finds weights to optimize the likelihood of the training data. Regularization is used to prevent overfitting, with the hyperparameter strength determined by "C". This gave better results than both Naïve Bayes and SGD classifier for top_category_id when "C" was set to 0.1. for color_id it gave almost same results as SGD classifier but not more than that.

At last, we used RandomForestClassifier from scikit-learn library. Random Forest was used as it is an ensemble learning algorithm that is commonly used for classification and regression tasks. It represents the mean prediction of all possible trees. Like SGDClassifier, it has hyperparameters which we adjusted such as "n_estimators=100", "random_state=42" to get good results. For top_category_id, it gave best results as compared to rest three algorithms as shown in table 1. And for color_id also it gave good results as shown in table 2.

5. Experiments and Results

5.1 Experiment

Different combinations of columns are used to train the data

and finalize which columns would be perfect to achieve better results. Hence, “title”, “keywords” and “tags” are the best columns to take as input for target attribute “top_category_id”. “title” column was cleaned properly, “tags” null values were imputed from “description” and “title” column and finally “keywords” column was obtained by doing proper feature engineering. All of this led to maximum results with Random Forest algorithm as shown in table 1.

For target attribute “color_id”, used only two columns “extracted_colors” and “first_occurrence_color” as input because all the required color words are extracted into these two columns from “title” and “description”. Also, there is a benefit of reduced computational time because no need to use extra columns. Feature extraction is done and created these two columns “extracted_colors” and “first_occurrence_color” as explained in the section feature engineering above. The linear SGDClassifier gave best results for “color_id” as shown in table 2. Hyperparameters for SGDClassifier were set as following to achieve these results-
loss='hinge', penalty='l2', alpha=1e-3, random_state=42, max_iter=5.

“Bottom_category_id” contains more than 2700 unique values so it was giving runtime error while executing. But was able execute two algorithms for this target attribute- linear SGDClassifier and MultinomialNB. Three input columns were used for this which were “tags”, “type” and “keywords”. These columns contain enough important information to proceed with the above two algorithms. Naïve Bayes gave better results as compared to linear SGDClassifier as shown in table 3.

Comparison is done amongst these four models to finalize which model will be best suited for the target features.

5.2 Results

The evaluation of Esty train data is shown in table 1 for top_category_id.

Table 1. top_category_id f1 score

Models	Precision	Recall	F1 score
Naive Bayes	0.75	0.65	0.6072
SGD classifier	0.77	0.76	0.7490
Logistic Regression	0.79	0.78	0.7777
Random forest	0.83	0.83	0.8225

The evaluation of Esty train data is shown in table 2 for color_id.

Table 2. color_id f1 score

Models	Precision	Recall	F1 Score
Naïve Bayes	0.74	0.73	0.7275
SGD Classifier	0.74	0.73	0.7369
Logistic Regression	0.74	0.73	0.7301
Random forest	0.74	0.73	0.7345

The evaluation of Esty train data is shown in table 3 for bottom_category_id.

Table 3. bottom_category_id f1 score

Models	Precision	Recall	F1 score
Naive Bayes	0.57	0.51	0.4873
SGD classifier	0.54	0.49	0.4766

6. Conclusion

In this paper, firstly we analysed the data and decided which columns are required to go forward with. Then did text pre-processing and feature engineering as mentioned in above section. Finally, we performed multi class classification on train data and got best f1 score for “color_id” with SGD classifier of linear regression, f1 score for “top_category_id” with Random Forest algorithm and f1 score for “bottom_category_id” with Naïve Bayes. This are the final algorithms for respective attributes as they gave maximum outcome. We don’t know the unseen test data; hence we have f1 scores for train data. Ppredicted all three attributes from provided test data with trained models and extracted that prediction in parquet file.

Further work on this includes identifying appropriate computational resources to guarantee that the models can operate successfully within a certain timeframe. Moreover, using tfrecords files, which contain images, could lead to more precise findings.

7. REFERENCES

[1] Sleiman, R., Tran, K.P. and Thomassey, S., 2022, July. Natural Language Processing for Fashion Trends Detection. In 2022 International Conference on Electrical, Computer and Energy Technologies (ICECET) (pp. 1-6). IEEE.

[2] Liu, C.Z., Sheng, Y.X., Wei, Z.Q. and Yang, Y.Q., 2018, August. Research of text classification based on improved TF-IDF algorithm. In 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE) (pp. 218-222). IEEE.

[3] Dubey, Y., Wankhede, P., Borkar, T., Borkar, A. and Mitra, K., 2021, December. Diabetes Prediction and Classification using Machine Learning Algorithms. In 2021 IEEE International Conference on Biomedical Engineering, Computer and Information Technology for Health (BECITHCON) (pp. 60-63). IEEE.

[4] Vasu, V.N., 2022, November. Prediction of Defective Products Using Logistic Regression Algorithm against Linear Regression Algorithm for Better Accuracy. In 2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT) (pp. 161-166). IEEE.

[5] JUGRAN, S., KUMAR, A., TYAGI, B.S. and ANAND, V., 2021, March. Extractive automatic text summarization using SpaCy in Python & NLP. In 2021 International conference on advance computing and innovative technologies in engineering (ICACITE) (pp. 582-585). IEEE.

[6] Berko, A., Matseliukh, Y., Ivaniv, Y., Chyrun, L. and Schuchmann, V., 2021, September. The text classification based on Big Data analysis for keyword definition using stemming. In 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT) (Vol. 1, pp. 184-188). IEEE.