Binary Classification to predict product matches using K-Means Clustering and Naïve Bayes Classifier

Bhavesh Bhagria

Student Id: 21262891

Abstract: Many retail websites sell products from many companies. To keep beating the competition, it becomes necessary for a company to keep track of its progress. Because e-commerce websites sell the same products under different names, it becomes a tedious task for a human to physically match every product. This problem requires a technical solution, that is fast, efficient, and is as accurate as a human's judgment. In this paper, I propose such a solution. The dataset contains three parquet files for training, testing, and matches, generated by humans, which require preprocessing. As this will be a binary classification task, into match and non-match. Then I will use the K-means clustering approach to understand the nature of clusters, and the Naïve Bayes classifier for final prediction.

CCS Concepts: • Computing Methodologies → Machine Learning, Modeling, and Simulation;

• Theory of computation → theory and algorithms for application domain;

• Mathematics of Computing → Probability and statistics.

Additional Key Words and Phrases: datasets, data manipulation, feature engineering, prediction.

1 INTRODUCTION

As a customer proposition, Zalando strives for "trustworthy" prices. That is, the company wants to offer competitive prices in each of its dynamic market environments, to alleviate its customers from having to compare prices and to drive revenue growth. To do that for its hundreds of thousands of individual products, Zalando needs to Identify exact product matches across the relevant European competitors. As one of the biggest e-commerce websites, Zalando must keep track of its competition. One of such companies is a rapidly growing company, AboutYou. Zalando wants to find out what same products is AboutYou also selling on its website. As there are millions of products on both websites, it is not feasible for a human to match all products. Using features such as brand, color, and price, we can use classification algorithms to predict whether a given set of potential matches are a match or not.

In this project, we will use such algorithms and perform data manipulation and feature engineering to make sure that the matches are accurate. Using K-Means clustering to understand the separation of matches and non-matches, and hopefully use it to make predictions based on clusters. Then I plan to use a classification algorithm to make final predictions.

In the coming sections, I will explain technically how I have executed the project and explain why I have used specific technologies.

2 LITERATURE REVIEW

[1] Müller, A.C. and Guido, S., 2016. *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc.".

In this book, Muller and Guido give detailed information applications of Machine Learning in python using the library sci-kit learn. It contains detailed information on how these algorithms are when to use them, and how to use them using the sci-kit learn. I found this book useful as I was using unsupervised learning algorithms for the first time, and the reasons I have given for my explanations below have been referenced from this book.

[2] McKinney, W., 2012. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. " O'Reilly Media, Inc.".

In this book, McKinney explains how to manipulate data in Python using Pandas and Numpy library. This book has been useful for me to study the commands that are necessary for performing pre-processing. I also found commands that helped to generate potential matches for my test data frame

[3] Ma, J., Jiang, X., Fan, A. *et al.* Image Matching from Handcrafted to Deep Features: A Survey. *Int J Comput Vis* **129,** 23–79 (2021).

This paper explains the nature of image matching. It talks about how image matching is done in high-dimensional data. As this project also has the features of images, this paper helped me understand how images can be used for product matching. The use of deep learning for feature-based matching can prove useful for similar projects in the future.

3 METHODOLOGY

The methodology includes exploratory data analysis, pre-processing, feature engineering, and building a machine learning model

Proposed Methodology:

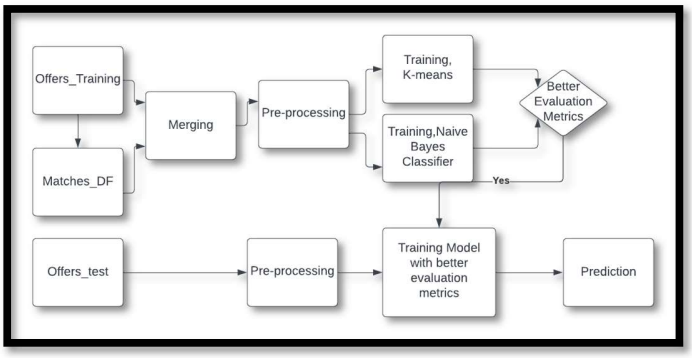


Fig. 1: Flowchart

i. Exploratory Data Analysis:

The dataset provided contains three parquet files. 'offers_training.parquet', which contains all products from Zalando and AboutYou, along with important features such as title, brand, color, textual description, and price of the product. 'matches_training.parquet' contains pairs of Zalando and AboutYou products that match this data frame was created by a human and this dataset plays a major role in creating the training data frame. The offers test file contains similar features after exploring the dataset and quantitative details about the products.

ii. Pre-processing and Feature Engineering

The offers training and offers testing dataset contains a lot of features that are necessary for making predictions, but these data frames are not in a format that is fit for training and testing. First, the details of all the Zalando are in the German language, so I translated them to English to have accurate similarity scores. For the offers training data frame, I only translated the description feature.

a. Training Dataframe.

For the training data frame, I used the matches training data frame. By using Boolean masking[2], I merged the matched products with the offers training to get all the features for matched products to get the numerical features (X features) and target variable (y feature) [1]. As I am using the title and description of products from both websites to generate similarity scores, I used the partial ratio function from the fuzzy-wuzzy library because it generates similarity scores based on certain words in a paragraph and not the paragraph as a whole [8]. After this process, I created a new data frame that had the offer ids, and similarity scores between their titles and descriptions as metrics for training.

I created a few pairs of no-matched products because only matches are not sufficient to train a model. The model also needs to know what kind of similarity scores do non-matched pairs have [1]. For this purpose, I removed the matched products from the offers training and found potential non-matches from that data frame. Then I followed the same procedure for generating similarity scores between their titles and descriptions. I chose 7500 pairs of non-matches (half the number of the matched pairs) to create. After that I concatenated both data frames to create a training data frame with matched and non-matched products, and an added column called 'status', which will be our y-variable, mapping 1 to matched products and 0 to non-matched products [1][2][4].

b. Testing Dataframe

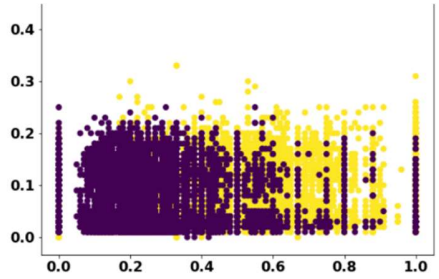
For testing the data frame, I first translated the color, title, and description feature to English. As brand and color can have the same values without any possible differentiation, I used brand and color as a filter for finding potential matches. By using the merge function with the filter of brand and color, I got 63 million potential matches [2]. After that, I generated similarity scores for the title and description of the potential matches [9]. By using random 25% of the potential matches, I created my testing dataset.

	zal_offer_id	ay_offer_id	fuzz_score_description	fuzz_score_title
0	37d0d991-4264-4dce-9980-17bde9d7b984	f76f3710-6c7f-4dc9-b9f1-0dbad109315d	0.01	1.0
1	a81b5fe2-97e3-48db-a9cc-782e7ad0154a	f76f3710-6c7f-4dc9-b9f1-0dbad109315d	0.02	1.0

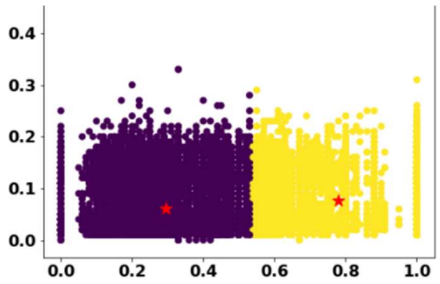
Fig. 2: Test Dataframe

iii. Machine Learning Model

For the machine learning model, I have created two models, however, I have used only one to make final predictions. As this is mainly a classification task, I wanted to use algorithms that are meant for classifications [2][3][5]. Among supervised learning algorithms, there are logistic regression and Naïve Bayes classifiers, and among unsupervised learning algorithms, there are various clustering algorithms, with K-Means clustering being the most popular[1]. I first decided to use K-Means clustering as I thought the two different clusters of matched and non-matched products would be appropriate for prediction[4].



This image depicts the relation between the data.



This image shows the generated clusters with generated centroids for both clusters.

Fig. 3 Clusters

As we can see in the above two pictures, there is a huge overlap between the match and non-match products, which has been adjusted after the generation of clusters. Therefore, I decided to use a different algorithm[7].

I used Naïve Bayes model, with the Gaussian version. The reason I chose to move ahead with Naïve Bayes is that it looks at all the parameters individually, assuming that they are not related or dependent on each other[1][5]. As the similarity scores for title and description are not dependent on each other, Naïve Bayes was the most appropriate choice for building this model. I used Gaussian NB to build my model because Gaussian NB can be applied to any continuous model [1][6]. Gaussian NB is mostly used with very high-dimensional data, but I thought our case was more than sufficient for the use of Naïve Bayes, as it is very fast to train and predict when compared to other linear models[1][5].

4 RESULTS AND DISCUSSION

After training the data, I generated my evaluation metrics, accuracy score, precision score, and recall score as shown below, with an overall accuracy score of 85%, a precision score of 90%, and recall score of 86%, Naïve Bayes classifier performed great in classifying the product pairs into match and non-match [1].

	precision	recall	f1-score	support
0	0.75	0.81	0.78	1506
1	0.90	0.86	0.88	3029
accuracy			0.85	4535
macro avg	0.83	0.84	0.83	4535
weighted avg	0.85	0.85	0.85	4535

Fig.4: Classification Matrix

The confusion matrix below shows that the model generated only 27 % true negatives, which means 27% of the data that was declared a non-match, was a non-match[1].

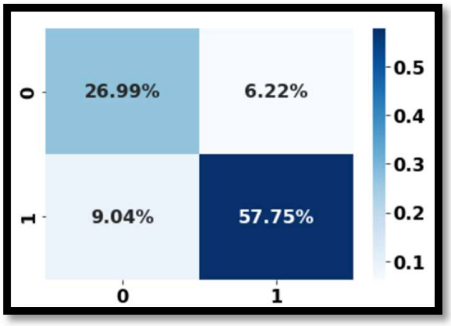


Fig. 5: Confusion Matrix

It also shows that the model generated only 58% True Positives, which means that the products that were declared a match, were a match[1].

Product from zalando | Product from AboutYou

5 CONCLUSION

In this project, I have manipulated the training and testing data frames a lot to suit the needs of the training model. After generating the appropriate training and testing datasets, I used K-Means clustering to train and predict the dataset, but we saw how the overlap in the dataset created restrictive clusters which could have generated many false positives and/or false negatives.

Using the Naïve Bayes classifier gave us a really strong machine learning model, which evaluation metrics above 85% and a great 58 % True Positive rate. For future work with similar datasets, images of the products can be accessed and used to predict matches by denoising the images, auto-sizing them to the same dimensions, and using similarity metrics to compare them and predict matches. Using images to predict matches by utilizing brand and color as a filter to find potential matches.

6 REFERENCES

[1] Müller, A.C. and Guido, S., 2016. *Introduction to machine learning with Python: a guide for data scientists*. " O'Reilly Media, Inc.".

[2] McKinney, W., 2012. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. " O'Reilly Media, Inc.".

[3] Ma, J., Jiang, X., Fan, A. *et al.* Image Matching from Handcrafted to Deep Features: A Survey. *Int J Comput Vis* **129,** 23–79 (2021). https://doi.org/10.1007/s11263-020-01359-2

[4] https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a

[5] https://towardsdatascience.com/all-about-naive-bayes-8e13cef044cf

[6] https://towardsdatascience.com/naive-bayes-classifier-explained-50f9723571ed

[7] https://www.kaggle.com/code/akshatpathak/text-data-clustering/notebook

[8] https://towardsdatascience.com/string-matching-with-fuzzywuzzy-e982c61f8a84

[9] https://towardsdatascience.com/17-types-of-similarity-and-dissimilarity-measures-used-in-data-science-3eb914d2681?gi=a560a5d656c4#:~:text=In%20data%20science%2C%20the%20similarity,are%20grouped%20into%20one%20cluster.