Amazon Review Classification and Sentiment Analysis

Nadi hanane

National School of Applied Sciences Al Hoceima

Hanane.nadi@etu.uae.ac.ma

Abstract

As e-commerce rapidly evolves, especially with the rise of cash-on-delivery networks, online customer reviews play an important role in shaping purchasing decisions. The feelings expressed by customers greatly influence how sellers choose and select products, offering valuable guidance for informed decisions.

This research offers an analysis of the Amazon reviews dataset, focusing on sentiment classification through various machine learning approaches. Initially, the reviews were converted into vector representation using the TF-IDF technique. Subsequently, the performance of diverse machine learning algorithms, encompassing logistic regression, naive Bayes, XGBoost, and SVM, was assessed using metrics such as accuracy, F1-score, precision, and recall. Then, we analyzed the best performance model in order to investigate its sentiment classification.

I. Introduction

In the expansive realm of online shopping, where choices abound, picking the right products has become a challenging and time-consuming task, particularly for sellers utilizing cash-on-delivery (COD) networks. The inherent risk of financial loss intensifies with the complexity of the manual selection process. Traditional approaches demand a substantial investment of time and resources, prompting the necessity for a smarter and more efficient alternative. This research addresses these challenges by introducing a methodology enriched with sentiment analysis. This approach not only aims to streamline and expedite the product selection process but also caters to the unique difficulties faced by buyers, especially those involved in cash-on-delivery transactions. Sentiment analysis, integrated with machine learning, acts as the linchpin, offering insights into consumer emotions and preferences to enhance the precision and efficacy of decision-making in the dynamic landscape of e-commerce.

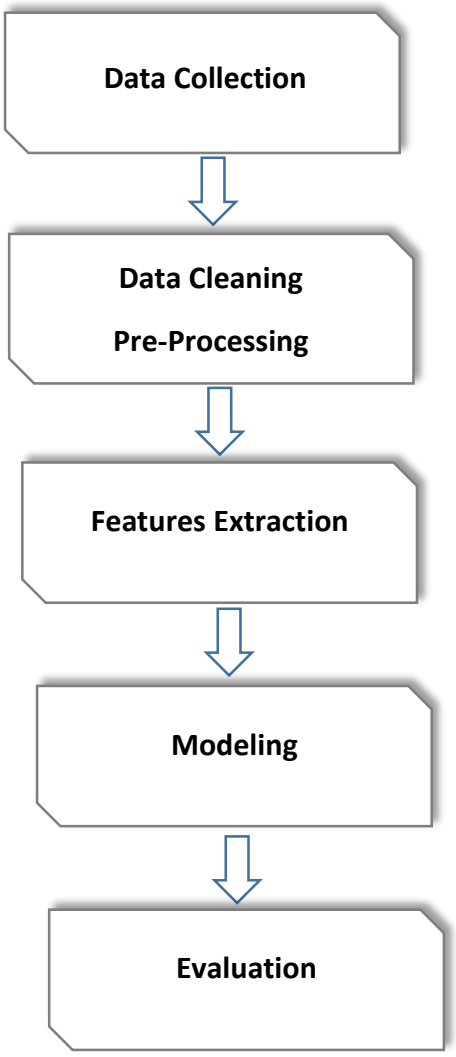
II. Related work

Various works in the literature have focused on the problem of identifying users' opinions of different products using Amazon reviews. Rathor, Agarwal, and Dimri [1] conducted a study focusing specifically on the 'iPhone brand, assessing the validity of algorithms in classifying online reviews. Conversely, Bansal and Srivastava [2] employed a supervised model in their research, aggregating 400,000 reviews from various brand names. Rathor, Agarwal, and Dimri processed data by not considering emoticon expressions, focusing on unique reviewer IDs, and applying feature reduction to eliminate stop words. In contrast, Bansal and Srivastava utilized the spaCy library for data cleaning, incorporating stemming, removal of stop words, and conversion to lowercase. Both

studies employed machine learning algorithms such as naive Bayes (NB) and support vector machine (SVM), with Bansal and Srivastava additionally incorporating logistic regression (LR) and random forest (RF). Rathor, Agarwal, and Dimri achieved the highest accuracy (81.20%) with SVM and weighted unigram, while Bansal and Srivastava reported RF with CBOW features achieving the highest accuracy (90.66%).

III. METHODOLOGY

This section provides an overview of the methodology and techniques employed for classifying Amazon Fashion reviews, aligning with common practices in the field of sentiment analysis. The following paragraphs detail the steps undertaken during the experiments. The figure below visually illustrates the sequential phases of this study, starting with the data collection until evaluating each classification model.



1. Data collection :

The data used in this study is a set of approximately 880,000 product reviews collected from Amazon.com by Julian McAuley [3] each review include information on rating, product id, helpful, reviewer id, review title, review time, and review text. The rating is based on a 5-star scale. Figure 1 displays a review in the dataset.


```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "helpful": [2, 3],
  "reviewText": "I bought this for my husband who plays the piano.
He is having a wonderful time playing these old hymns.  The music  is
at times hard to read because we think the book was published for
singing from more than playing from.  Great purchase though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```


Figure 1: Example of an Amazon.com review

2. Data Cleaning and Pre-Processing

➢ Data Cleaning

In the process of getting our data ready for analysis, we take a few important steps. First, we make sure there aren't any identical copies of the same information, it's like cleaning up duplicates. We also get rid of things that won't help us, like columns with irrelevant details. Additionally, we check if there's any missing information and fix it.

➢ Pre-Processing

Data pre-processing plays an important role to enhance the quality of textual data. Figure 2 illustrates the various pre-processing steps that were implemented on the Amazon dataset for this research.

Each review in the dataset is assigned a label as positive or negative based on its star rating. To maintain consistency, all letters are converted to lowercase. Additionally, we eliminate punctuation and common stop words, like "a, an, with, etc.," that don't significantly contribute to meaning. Next, the reviews undergo tokenization, breaking sentences into a sequence of words known as "tokens." To ensure simplicity and uniformity, these tokens are then brought back to their root forms through lemmatization.

Note:
In sentiment analysis, removing stop words is standard, but caution is needed with negation words like "not." Simply discarding them may alter the sentiment. For instance, turning "I do not like this movie" into "like movie" changes the negative sentiment to positive. To address this, it's important to modify the stop words list to exclude negations to preserve the accurate sentiment context in text analysis.

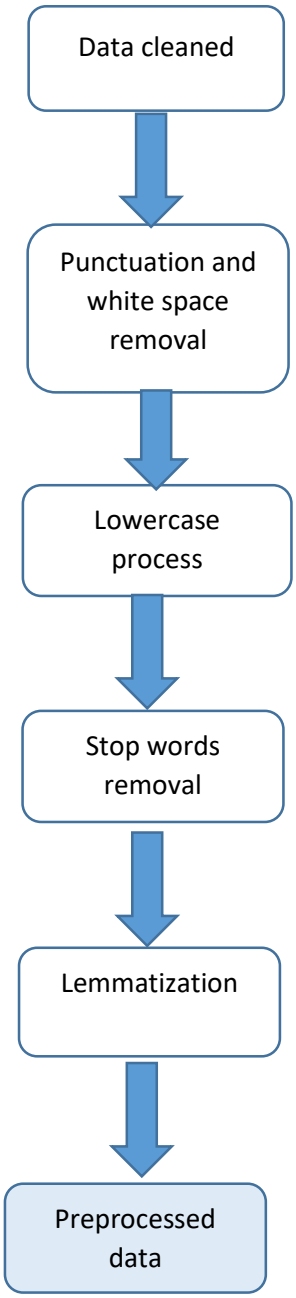


Figure 2: Pre-processing steps

Due to the substantial size of our dataset, we will extract a subset comprising only 30,000 reviews and to maintain balance, we will select instances where the positive class is equal to the negative class. This subset will be further divided, allocating 75% for training and 15% for testing purposes when constructing baseline models.

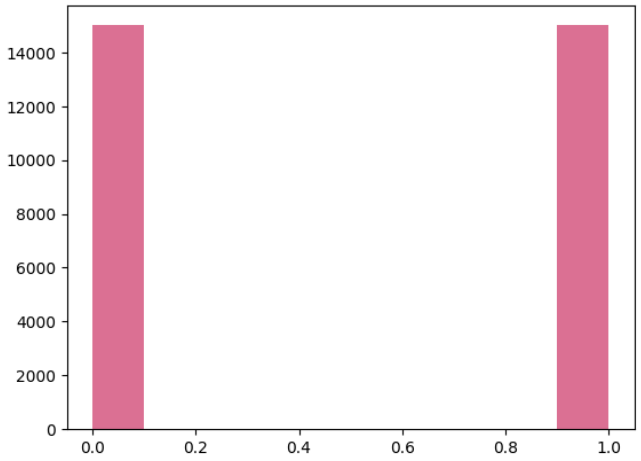


Figure 3: Distribution of Reviews by Rating

As shown in Figure 3, there are 15,000 positives reviews (class 1) and 15,000 negative reviews (class 0)

3. Features Extraction

In this study, we employed a comprehensive text processing methodology to convert fashion reviews into numerical vectors for sentiment analysis on Amazon. To achieve this, we employed the TF-IDF (Term Frequency-Inverse Document Frequency) technique.

Term Frequency (TF) counts how often a word shows up in a document, and Inverse Document Frequency (IDF) measures how unique a word is across the whole dataset.

$$\text{TF (t, d)} = \frac{Number\ of\ times\ t\ appears\ in\ d}{Total\ numbers\ of\ terms\ in\ d}$$

$$\text{IDF (t)} = \log \times \frac{Total\ number\ of\ documents}{Total\ document\ with\ the\ terme\ t}$$

In a document D, the final weight for a term t is calculated as:

$$\text{TF-IDF} = TF(t, d) \times IDF(t)$$

By multiplying TF with IDF, we obtained numerical weights for words, indicating their importance in a document within the corpus.

Additionally, we incorporated n-grams, examining word sequences from single words to sets of three, to capture the contextual nuances of language and better understand the sentiments expressed. By combining TF-IDF and n-grams, our approach not only identifies key words but also considers how words work together, facilitating a more detailed and straightforward sentiment analysis of fashion reviews on Amazon.

IV. Models

To investigate which machine learning modality performs best on the classification of Amazon fashion reviews, four different machine learning modalities were trained.

1. Logistic Regression

Logistic Regression is a classification algorithm that predicts the probability of an instance belonging to a particular class. It utilizes the logistic function to map any real-valued number into a value between 0 and 1, making it suitable for binary classification tasks.

$$f(x) = \frac{1}{1+e^{-x}}$$

2. Multinomial Naive Bayes

Multinomial Naive Bayes is a probabilistic classification algorithm relying on Bayes' theorem with an assumption of feature independence. It is well-suited for tasks like text classification, where the frequency of words is crucial. The algorithm calculates the probability of a class given observed features, using Bayes' theorem to update the probability estimates based on new evidence.

$$P(A|B) = P(A) * P(B|A)/P(B)$$

More generally:

$$P(x_1,\ldots, x_k|y) = \prod_{i=1}^{k} p(x_i|y)$$

3. Extreme Gradient Boosting (XG boost)

XGBoost is a powerful ensemble learning algorithm that builds decision trees sequentially to correct errors. It builds a series of decision trees sequentially, each correcting errors made by the previous one. XGBoost combines the predictions from multiple weak learners, creating a robust and accurate model. It uses a gradient boosting framework, optimizing the overall performance by minimizing a loss function

4. Support Vector Machines (SVM)

Support Vector Machine (SVM) is a powerful classification algorithm that works by finding the hyper plane that best separates data points of different classes. SVM aims to maximize the margin between classes, making it effective in high-dimensional spaces and particularly useful in scenarios with complex decision boundaries. The optimization problem that SVM tried to solve is below:

$$\arg max_{\gamma,\omega,b} \frac{1}{2} ||W||^2$$

$$\text{s.t.} y^i (w^T x + b) \geq 1, \; i = 1, 2, \ldots m$$

V. Results and discussion

1. Evaluation metrics

In the process of determining the most accurate classification algorithm for our test set, we utilized critical performance metrics, including accuracy, precision, recall, and F1-score. These metrics are essential for assessing the effectiveness of supervised machine learning algorithms, and their computation relies on information derived from the confusion matrix.

- **Confusion matrix.**

The confusion matrix is a crucial evaluation tool that breaks down a model's performance into counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These elements help compare class labels, distinguishing between correct and incorrect predictions. True Positives are instances correctly identified as positive, while False Positives are predicted positives that are actually negative. Similarly, True Negatives are correctly identified negatives, and False Negatives are predicted negatives that are actually positive. Precision, recall, F-measure, and accuracy are derived from the confusion matrix, offering key metrics to assess the overall performance of classifiers.

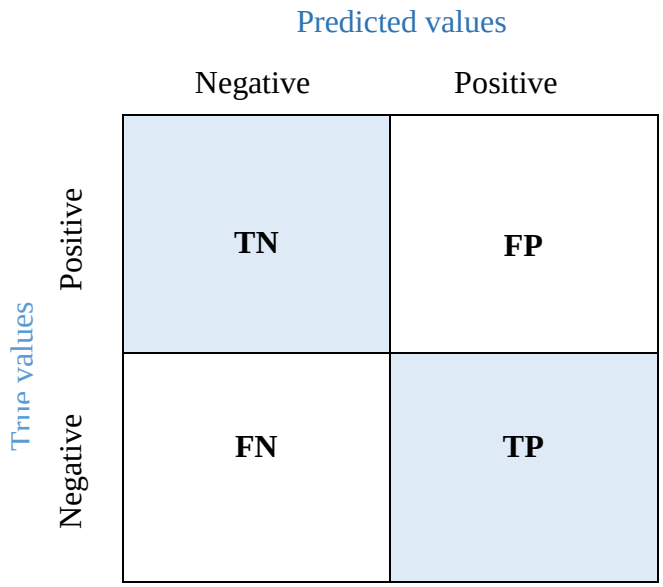


Figure 4: Confusion matrix

- **Accuracy :**

Accuracy is defined as the percentage of reviews that are classified correctly divided by the total number of reviews.

$$Accuracy = \frac{TN + TP}{TN + FP + FN + TP}$$

- **Precision**

Precision measures the percentage of positive reviews that predict truly divided by the total number of reviews that are classified positive.

$$Precision = \frac{TP}{TP + FP}$$

- **Recall** :

Recall measures the percentage of the reviews that classify positively divided by the total number of reviews which are truly positive.

$$Recall = \frac{TP}{TP + FN}$$

- **F1-score**

F1-score combines both precision and recall

$$F1 - score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

2. Comparison of different sentiment analysis models

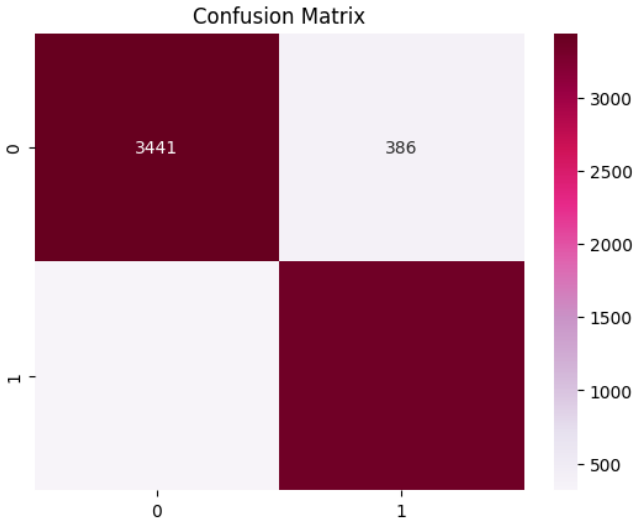


Figure 5: Confusion matrix of logistic regression

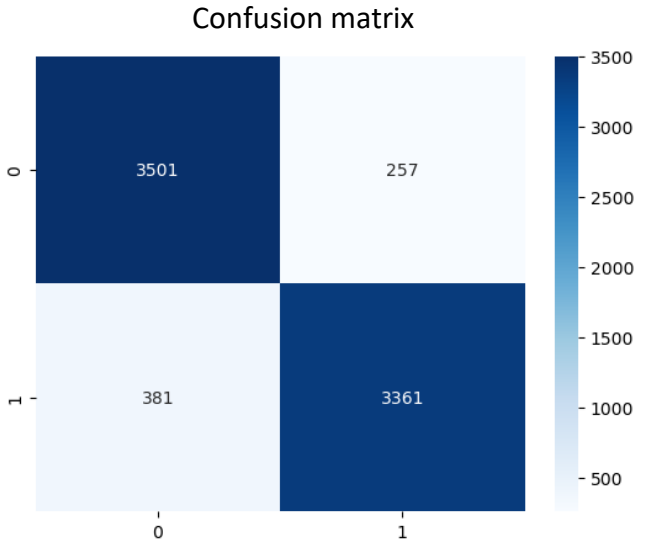


Figure 7: Confusion matrix of naive bayes

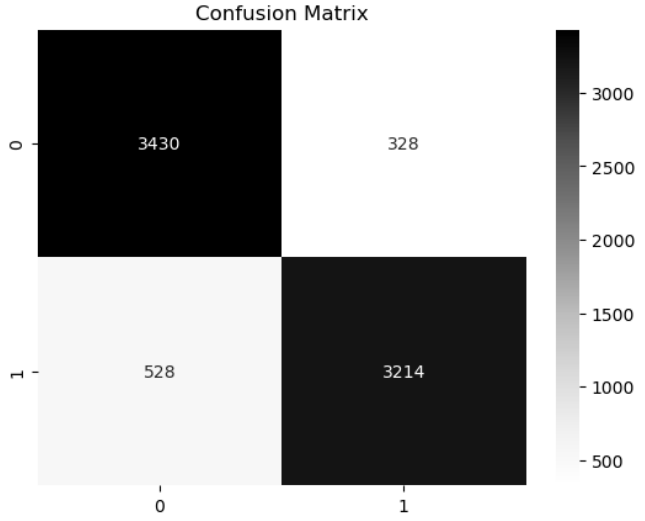


Figure 8: Confusion matrix of XGboost

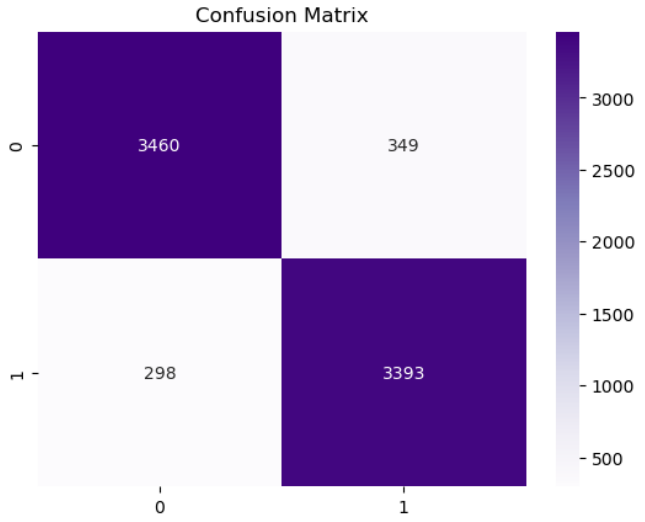


Figure 9: Confusion matrix of SVM

	Accuracy	Precision	Recall	F1-score
LG	0.9062	0.9136	0.8968	0.9051
MNB	**0.9149**	**0.9289**	0.8981	**0.9133**
XGB	0.8858	0.9073	0.8588	0.8824
SVM	0.9137	0.9192	**0.9067**	0.9129

Table1: result before hyperparametres tuning

In reviewing the confusion matrices and performance metrics, Naive Bayes emerges as the top performer with the highest accuracy (91.49%), precision (92.89%), recall (89.81%), and F1-score (91.33%). Following closely are Support Vector Machine (SVM) and Logistic Regression, both exhibiting competitive results. SVM achieves an accuracy of 91.37%, precision of 91.92%, and recall of 90.67%, while Logistic Regression attains 90.62%, 91.36%, and 89.68% for these metrics, respectively. Despite slightly lower scores, XG Boost still delivers respectable results, featuring an accuracy of 88.58%, precision of 90.73%, recall of 85.88%, and an F1-score of 88.24%.

In conclusion, Naive Bayes consistently performs well across all metrics, making it a strong candidate for sentiment analysis on fashion reviews. SVM and Logistic Regression also demonstrate competitive performance, while XG Boost, although slightly lower, still provides respectable results.

3. Improvement of models by hyper-parameters tuning

Fine-tuning hyper-parameters is a crucial step in crafting effective machine learning models. It's about adjusting the external settings of algorithms to make them work best for a specific dataset. In our project, we used this fine-tuning approach for all four algorithms to see how it affects each one's performance.

We fine-tuned the models using GridSearchCV from scikit-Learn, exploring different combinations of settings in a systematic way. This method checked each setup through cross-validation, giving us accuracy and other metrics to figure out the best configurations. The refined results show that this careful fine-tuning has a positive impact, making our machine learning models perform even better overall.

	Accuracy	Precision	Recall	F1-score
LG	0.9138	**0.9181**	0.9083	0.9132
MNB	0.8858	0.9073	0.8588	0.8824
XGB	0.8873	0.9040	0.8661	0.8846
SVM	**0.9168**	0.9175	**0.9155**	**0.9165**

Table2: result after hyperparametres tuning

We can see that Logistic Regression exhibits improvements in precision, recall, and F1-score, achieving an accuracy of 91.38%. Naive Bayes maintains consistency in metrics with an accuracy of 88.58%. XG Boost shows a slight enhancement in precision, recall, and F1-score, reaching an accuracy of 88.73%. Support Vector Machine (SVM) demonstrates notable improvements across the board, achieving an accuracy of 91.68%. With these refinements, **the SVM model emerges as the best-performing**, offering enhanced precision, recall, and F1-score for sentiment analysis on fashion reviews.

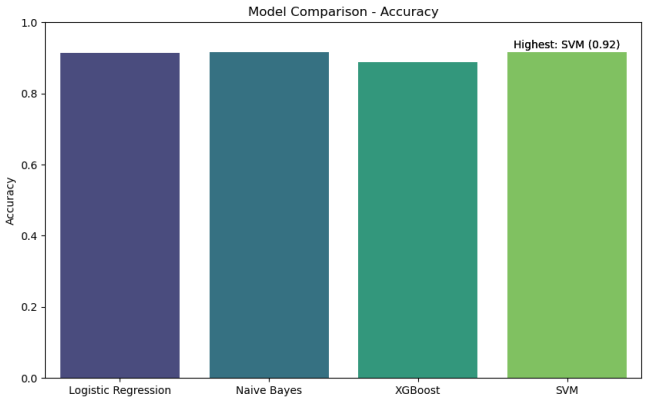


Figure 10: Model comparison

VI. Deployment

In the deployment phase of the project, I crafted a user-friendly website employing HTML and CSS for an appealing and intuitive design. This website serves as an interactive platform for a comprehensive understanding of the project's insights. Users can conveniently input the URL of an Amazon product via a straightforward mechanism.

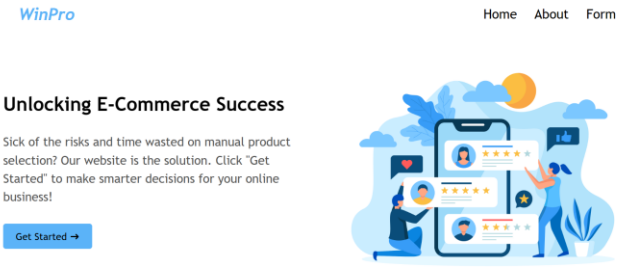


Figure 11: Home page of the website

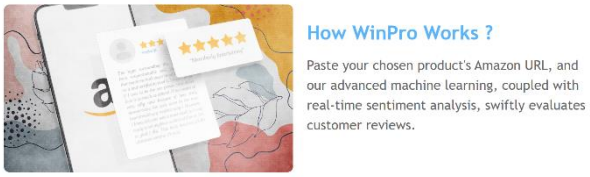


Figure 12: About page of the website

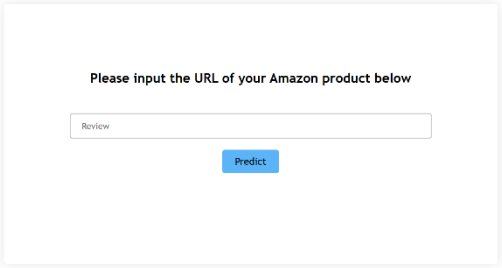


Figure 11: Form page of the website

Behind the scenes, the website utilizes web scraping techniques to gather reviews associated with the specified product. The collected data then undergoes meticulous cleaning and preprocessing to ensure optimal quality for subsequent analysis.

Once the data is prepared, the deployed machine learning model, integrated with Flask, comes into play. It predicts the outcome, determining whether the product is likely to be successful or not based on the analysis of positive and negative reviews. This seamless integration of technologies provides users with a quick and informative assessment of the product's sentiment and potential market success.

VII. Limitations

The project has several limitations. Firstly, its predictive accuracy is constrained by the need for a large and diverse dataset, as the model was specifically trained on fashion-related data. It may not perform as effectively on reviews from different product categories. Secondly, the model may struggle with interpreting language nuances, such as polarity, negation, and sarcasm, affecting sentiment analysis accuracy. Additionally, the project currently overlooks the influence of product pricing on consumer sentiment, which is a crucial factor in determining a winning product. The reliance on web scraping introduces potential vulnerabilities related to website changes. Lastly, the model's predictive capabilities may not promptly adapt to evolving consumer sentiments and emerging trends in online reviews.

VIII. Conclusion

In this project, we set out to analyze sentiments in Amazon fashion product reviews with a specific focus on predicting the winning product. Our approach involved using TF-IDF and n-grams to convert text into numerical vectors, enhancing our model's understanding of sentiments. The user-friendly website we developed allows users to input Amazon product URLs, initiating sentiment analysis that predicts a product's potential success based on positive and negative reviews. The project showcases the effectiveness of our methods in understanding sentiments and predicting successful fashion products.

References

[1] Rathor, A. S., Agarwal, A., & Dimri, P. (2018). Comparative study of machine learning approaches for Amazon reviews. *Procedia Computer Science*, 132, 1552–1561.

[2] Bansal, B., & Srivastava, S. (2018). Sentiment classification of online consumer reviews using word vector representations. *Procedia Computer Science*, 132, 1147–1153.

[3]https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2/

NLP_Final_Project.pdf (swarthmore.edu)

A-Comparison-of-Sentiment-Analysis-Methods-on-Amazon-Reviews-of-Mobile-Phones.pdf (researchgate.net)

Microsoft Word - 5463-JCS (researchgate.net)

https://youtu.be/QpzMWQvxXWk?si=pwToDnQKHg1OauEv