UT DALLAS
The University of Texas at Dallas

BUAN 6340 – Programming for Data Science

Course Project

Amazon Fine Food Review via Classification: Sentiment Analysis

- Final Report and Code

Group members:

Nayan Nandakishore Joshi (NXJ210028)

Nandita Vesangi (NXV220010)

Nathan Vaz (NXV220015)

Contents

Abstract:

The objective of this project was to perform sentiment analysis on Amazon Fine Food reviews using classification techniques.

The motivation behind this project was to perform sentiment analysis on Amazon Fine Food reviews. Sentiment analysis provides valuable insights into customer opinions and helps businesses understand customer satisfaction levels and improve their products or services. By analyzing the sentiment expressed in the reviews, the project aimed to extract meaningful information and classify the reviews as positive or negative. This classification can assist in making data-driven decisions, enhancing customer experience, and gaining a competitive advantage in the market.

The dataset contained a large number of reviews labeled as positive or negative.

To start, the dataset was preprocessed using natural language processing techniques, including removing duplicates and balancing the class distribution. The reviews were then transformed into numerical representations using Bag of Words (BOW) and TF-IDF methods.

Two machine learning algorithms logistic regression, and K-nearest neighbors (KNN), were selected for classification. The models were trained and evaluated using performance metrics such as accuracy, precision, recall, F1 score, and AUC-ROC.

Both logistic regression and KNN models were provided with inputs from both BOW and TF-IDF representations. The models were compared, and the one with the highest accuracy was selected as the final choice for predicting review sentiment.

Through the project "Amazon Fine Food Review via Classification: Sentiment Analysis," two key takeaways were:

1.Preprocessing Techniques: Preprocessing the dataset played a crucial role in improving model performance. Techniques such as removing duplicates and balancing the class distribution helped to ensure data quality and reduce bias. Additionally, applying NLP techniques like stemming and stop words removal helped to create more meaningful and concise representations of the text data, improving the efficiency and effectiveness of the models.

2.Model Selection and Evaluation: The choice of machine learning algorithms and feature representations significantly impacted the performance of the sentiment analysis task. Comparing multiple models, such as logistic regression and K-nearest neighbors, allowed for an informed decision based on accuracy and other evaluation metrics. Furthermore, using a variety of performance metrics, such as precision, recall, and F1 score, provided a comprehensive understanding of the models' strengths and weaknesses.

Introduction:

In today's digital age, the widespread use of social media and internet platforms has revolutionized the way companies across various industries market their products and services. The internet has effectively placed a vast array of information and resources at our fingertips, profoundly transforming the way people shop for goods and make purchasing decisions. From buying groceries to researching and purchasing a new car, virtually everything can be found and accessed on the internet from the comfort of one's own home.

The importance of social media in this digital landscape is significant, as it has emerged as a potent tool for customers to obtain information about products and services. Social media platforms provide a convenient and accessible medium for customers to discover, learn about, and evaluate different offerings. Customers now rely heavily on online customer evaluations and reviews, often valuing them more than the information offered directly by vendors.

Recognizing the significance of customer reviews, this project aims to develop a robust classification model to categorize customer reviews into two distinct categories: good and bad. By analyzing a dataset consisting of customer reviews and associated scores, we can train a machine learning model that can effectively determine whether a review is positive or negative. This classification model will enable companies to extract valuable insights from customer feedback, empowering them to make informed decisions and prioritize products with the most positive reviews.

By leveraging the power of machine learning and natural language processing techniques, this project offers companies a means to harness the wealth of customer reviews available online. The ability to accurately classify reviews as either good or bad provides a powerful tool for businesses to improve their understanding of customer preferences, enhance their product recommendations, and ultimately deliver a superior customer experience. Through this project, we aim to unlock the potential of customer reviews as a valuable resource for companies, enabling them to thrive in the digital era and meet the evolving needs of their customers.

Datasets and Preprocessing:

Amazon Fine Food Review- https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews

In this project, we will use a Kaggle dataset comprising product reviews. This data consists of reviews of fine food from amazon, over a period of more than 10 years up to October 2012. The dataset consists of 568454 observations with 10 variables (the Score, the Text, and other variables).

Because the dataset is too enormous, we elected to produce a smaller dataset of 10k observations using random selection.

In our analysis, we are primarily concerned with the textual review and forecast the score for that review. The score goes from 1 to 5, and it will be classified as either good (rating >= 3) or bad (rating 3). To conclude, we divide the review into two categories depending on its sentiment using logistic regression.

The objective of this project is to use a classification algorithm to distinguish the Score.

For the dataset mentioned above:

- We perform data cleaning i.e., removing null and duplicate values if any.
- Then, text preprocessing is applied wherein we remove junk words, special characters and the HTML tags that are present in the text that provide no meaning.

- Next, we perform stop words removal where the words that occur commonly across the documents in the corpus are removed.
- Now on the processed data obtained from the previous step, we perform stemming to bring similar meaning words to their roots form. Example: The words operate, operating, operated and operation have similar meaning, so can be brought down to their root form as ‘oper’.

Now we create bag of words for the above processed data. In this approach, we use the tokenized words for each observation and find out the frequency of each token and store it in a vector. The process of converting text into numbers is called vectorization. The following two methods can be used to perform this process:

- Count Vectorizer:

It works on term frequency i.e., counting the occurrence of each word in a document and creating a sparse matrix.

- TF-IDF (Term frequency inverse document frequency):

It scores the frequency of word in current document.

TF = Number of repetitions of a word in sentence / Number of words in the sentence

IDF = log (Total number of sentences / Number of sentences containing the word)

TF-IDFscore = TF*IDF

We will split this preprocessed data into training set (7k records) and testing set (3k records). We will use training set to train our model, while testing set will be used to evaluate the model. The algorithm that we’ll be using is Logistic Regression and K-Nearest Neighbor to classify the text into different categories of scores.

Since Count Vectorizer and TF-IDF will give out outputs that are not similar, we will train four models, two for Count Vectorizer (using Logistic Regression and KNN) and two for TF-IDF separately (using Logistic Regression and KNN).
Finally, the model that performs better and gives out the most accurate result will be considered as the final model.

Data Analysis:

In our dataset, we have observed that there are multiple reviews provided by a single user for a particular product ID. These duplicate records can be redundant and potentially impact the performance of the model. To address this issue, we have implemented a solution by sorting the values based on the product ID and then dropping duplicate values considering the subset of UserID, ProfileName, Time, and Text. This approach helps us handle duplicate reviews from a single user effectively.

Upon examining the "Score" column, which contains positive or negative values, we have found that the dataset is highly imbalanced. There are 307,063 positive values and only 57,110 negative values.
Imbalanced datasets can have several implications for a classification model, including bias towards the majority class, poor performance on minority classes, biased evaluation metrics, sensitivity to data distribution, and overfitting on the majority class.

To address the imbalance, we have employed the technique of oversampling. Specifically, we have selected an equal number of positive and negative values from the dataset, resulting in a balanced dataset of 10,000 positive and 10,000 negative values. By achieving a balanced dataset, we mitigate issues related to biased predictions, poor performance, improper evaluation metrics, and overfitting.

Next, we have created a bag-of-words representation based on the unprocessed text. However, the unprocessed text contains many irrelevant words that do not contribute to predicting the nature of the review. Additionally, training a dataset with such a large number of rows (over 10,000) and columns (around 30,000) would consume significant time and resources. To address these concerns, we have applied natural language processing (NLP) techniques such as stemming and stop word removal. These techniques help convert the data into a processed form that is smaller in size and more suitable for analysis.

By implementing these steps, we have successfully handled duplicate reviews, balanced the dataset, and processed the text data using NLP techniques. This approach ensures a more efficient and accurate analysis of the dataset, enabling us to build a robust classification model for predicting the sentiment of product reviews.

In general, Logistic Regression and KNN algorithms are sensitive to the scale of the input features. If the features have different scales, it can lead to biased results, as the algorithm might give more weight to the features with larger scales. Scaling the input data can help to mitigate this issue and improve the performance of these algorithms.

StandardScaler is commonly used for scaling the data in machine learning. It is a method for standardizing or normalizing numerical features in a dataset. Standardization involves transforming the data so that it has zero mean and unit variance.

It is not specifically a method for data analysis. It is typically applied as a step in the data preprocessing pipeline to prepare the data for analysis with machine learning algorithms. By scaling the data, StandardScaler can help improve the performance and convergence of various models, especially those that are sensitive to differences in feature scales, such as linear models or distance-based algorithms like K-nearest neighbors.

Models:

For our project, we have selected two machine learning algorithms: logistic regression and K-nearest neighbors (KNN). Logistic regression is a statistical model used for binary classification, while KNN is a proximity-based classification algorithm.

Logistic regression is a statistical model used for binary classification tasks. It predicts the probability of an event occurring based on input variables. It calculates a weighted sum of the input variables and applies a logistic function to transform the result into a probability score between 0 and 1.

K-nearest neighbors (KNN) is a machine learning algorithm that classifies or predicts based on the proximity of data points. It assigns the majority class among the K nearest neighbors for classification or uses their average/median value for regression. It is non-parametric, stores the training data, and is computationally expensive for large datasets. Choosing K and a distance metric are key considerations for KNN.

The primary focus of our problem statement is on determining whether a particular review will be good or negative. Consequently, this turns into a binary classification problem, which is why we favor classification techniques. Because of this, we use the logistic regression algorithm, a binary classifier, and the K closest neighbors' approach, a classification algorithm.

Since choosing K is a key consideration for KNN algorithm, we have created an odd list of K values ranging from 1 to 20. We performed 10-fold cross-validation, calculating the mean cross-validation scores for each value of K. By selecting the model with the lowest misclassification error, we identified the optimal K value. By employing both logistic regression and KNN algorithms, we aim to build effective classifiers for our project, enabling accurate prediction of positive and negative reviews.

In our project, we performed preprocessing on the reviews using natural language processing (NLP) techniques. After preprocessing, we utilized two feature representation methods: Bag of Words (BOW) and TF-IDF (Term Frequency-Inverse Document Frequency) on the cleaned text. We employed both BOW and TF-IDF inputs for our logistic regression and K-nearest neighbors (KNN) models.

To evaluate the performance of these models, we focused on accuracy, aiming to correctly classify the reviews as positive or negative. We compared the accuracy of all four models: logistic regression with BOW, logistic regression with TF-IDF, KNN with BOW, and KNN with TF-IDF.

Based on the accuracy results, we selected the model that achieved the highest accuracy as our final choice for predicting the output. This approach ensures that our model can effectively classify reviews as either positive or negative, providing accurate insights into sentiment analysis.

For the model KNN with BOW, we get an accuracy of 66.56%.

For the model Logistic Regression with BOW, we get an accuracy of 81.09%.

For the model KNN with TF-IDF, we get an accuracy of 60.9%.

For the model Logistic Regression with TF-IDF, we get an accuracy of 80.13%.

Among the available models, Logistic Regression with BOW demonstrates the highest level of accuracy, achieving an impressive 81.09% accuracy rate.

Results:

The result that we obtained from 10 observations of the testing datasets compared to the actual results are as follows:

Actual output: [1,0,1,0,0,0,0,1,1,1]

Predicted output: [1,0,0,0,0,1,1,1,1,1]

Our model has an approximate accuracy of 81.10%.

Since, we finalized the Logistic Regression model, the performance metrics that we considered is the accuracy which measures the overall correctness of the predictions, representing the proportion of correctly classified instances out of the total instances in the dataset.

The major drawback which is of a concern is that if the sentence sounds neutral but is not neutral. For instance, "the food was very good, but the packaging was not at its best" the model does get confused by giving more weightage to the packaging than the food's taste.

A solution for this type of problem is to use a more complex model which expertly handles sequence data. One of such models can be Recurrent Neural Network (RNN) which have connections that form a directed cycle, allowing them to maintain and process information from previous steps in the sequence.

References:

1. Amazon Fine Food Review- https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews
2. J. McAuley and J. Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. WWW, 2013.