Natural Language Processing

Customer Feedback Processing

Diego Rosa Paz
Sylvia Pérez Montero

January 31, 2025

Introduction

This business case outlines the development of an NLP model to automate the processing of customer feedback for a retail company.

Its goal is to evaluate how traditional ML solutions (NaiveBayes, SVM, RandomForest, etc) compares against a Deep Learning solution (e.g, a Transformer from HuggingFace) when trying to analyse a user review, in terms of its score (positive, negative or neutral).

Problem Statement

The company receives thousands of text reviews every month, making it challenging to manually categorize and analyze, and visualize them. An automated system can save time, reduce costs, and provide real-time insights into customer sentiment. Automatically classyfing a review as positive, negative or neutral is important, as often:

- Users don't leave a score, along with their review
- Different users cannot be compared (for one user, a 4 might be great, for another user a 4 means "not a 5" and it is actually bad)

Project Goals

The systems created in this project will be able to run classification of customers' reviews (the textual content of the reviews) into positive, neutral, or negative.

In addition, a comparison will be conducted to establish which solution yields better results, one that reads the text with a Language Model and classifies into "Positive", "Negative" or "Neutral", or one that transforms reviews into tabular data and classifies them using traditional Machine Learning techniques.

About the Data

The dataset used for this project consists of over 34,000 consumer reviews of Amazon products, provided by Datafiniti's Product Database. It includes basic product information, rating, and review text for each rating.

(source: https://www.kaggle.com/datasets/datafiniti/consumer-reviews-of-amazon-products/data)

Approach

A. Traditional NLP & ML Approach

a. Data Preprocessing

Loading of the dataset, cleaning, tokenization, lemmatization and vectorization was done prior to building the corresponding models. More specifically:

- removing irrelevant columns
- checking for empty and duplicate values
- lowercasing the text
- removing numbers
- removing punctuation
- removing extra spaces
- tokenization
- removing stopwords
- lemmatization (converts words to their base/root form)

b. Prior to building the models, the dataset was split into training and test data.

c. Model Building

i. Model selection - With respect to the machine learning algorithms for text classification, Naive Bayes, Logistic Regression, and Random Forest models were considered. BERT and DistilBERT were deep learning models considered as well within this project.

ii. Model Training - the selected models were trained on the preprocessed text data.

iii. Model Evaluation - several evaluation metrics were used to assess each model's performance: accuracy, precision, recall, F1 score and confusion matrix. The results were then used to compare each model and determine which performed the best.

B. Transformer Approach

a. **Data preprocessing** - same considerations as in the traditional approach.

b. Model building

i. **model selection** - Bidirectional Encoder Representations from Transformers (BERT) and DistilBERT (a lightweight version of BERT) were used for their utility for text classification tasks. BERT was selected for using with AWS because of its power requirements and its recommended use for research and large scale applications. DistilBERT was used based on it being a lighter and faster version of BERT.

ii. **model training** - same as in the traditional approach.

iii. **model evaluation** - same as in the traditional approach.

C. Performance Analysis & Comparison

a. Each model was evaluated individually and in comparison, to the other models.

Results

Model	Model Accuracy	F1 Score	Precision	Recall
DistilBERT	**93.37%**	**90.17%**	**87.18%**	**93.37%**
Support Vector(TF-IDF)	71.77%	83.00%	73.00%	97.00%
Random Forest (TF-IDF)	71.61%	83.00%	73.00%	97.00%
Random Forest (Count)	15.52%	9.00%	21.00%	19.00%
Naïve Bayes (Count)	11.90%	9.00%	22.00%	18.00%
Naïve Bayes (TF-IDF)	11.90%	9.00%	22.00%	18.00%
Logistic Regression (TF-IDF)	6.19%	5.00%	21.00%	16.00%
Logistic Regression (Count)	6.19%	5.00%	21.00%	16.00%

- DistilBERT had significantly better results than the traditional Naïve Bayes, Logistic Regression, Random Forest and Support Vector models.
    - **Accuracy** of 93.37% was dramatically higher

    - **F1-Score** of 90.17% suggests a strong balance between precision and recall, whereas the traditional models had significantly values
    - **Precision** (87.18%) and **recall** (93.37%) show that DistilBERT predicts very well and also captures most relevant cases.
- While the traditional models struggled with class imbalance (even with tuning), DistilBERT appears to generalize well.
- Within the traditional models, Random Forest outperformed Naïve Bayes and Logistic Regression, but still had much lower accuracy than DistilBERT.

Optimization Recommendations

As with other modeling exercises, with more time and resources further optimization should be explored. Among the recommendations we feel necessary are:

- Increase training epochs for DistilBERT (5-10 versus the 3 used here)
- Hyperparameter tuning:
    - Batch size (8 ->16)
    - Learning rate
    - Max length of sequences (256 -> 512)
- Look further into adjusting class weights, particularly of the 1 and 2 ratings.
- Combine DistilBERT with other Transformers