



Customer Reviews Sentiment Analysis

Caroline Araujo, Dani Siaj, Saul Verdu

Project Overview:

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

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

Content:

This is a list of over 34,000 consumer reviews for Amazon products like the Kindle, Fire TV Stick, and more provided by Datafiniti's Product Database. The dataset includes basic product information, rating, review text, and more for each product.

Acknowledge:

This dataset is a partial sample of a large dataset

Exploratory Data Analysis

Original Shape of Database: 34660 Rows, 21 Columns

Final Shape: 346266 Rows, 3 Columns

Dataset Not Clean: 34 missing values after Dataframe transformation.

High imbalance in the amount of data for each rating (most of the reviews are 5.0, only 2208 reviews where negative)

The values in column 'reviews.rating' are floats: 1.0, 2.0, 3.0, 4.0, 5.0

Modifications:

Dataframe transformation: Only kept 3 columns (['categories', 'reviews.rating', 'reviews.text'])

Values in column 'reviews.rating' are replaced with 'positive' for 5.0, 'neutral' for 4.0 and 'negative' for 1.0, 2.0, 3.0 for standardization

Balanced dataframe was built with a similar count of data points for each rating, preventing a biased model later.

Model Architecture

Model Comparison:

We compare traditional prediction models with pre-trained models from 'Hugging Face Models:

- Multinomial NB
- Logistic Regression
- SVC
- Random Forest
- Pipeline ('sentiment analysis', 'cardiffnlp/twitter-roberta-base-sentiment')
- Fine tuning of Pipeline, 'cardiffnlp/twitter-roberta-base-sentiment'

Model Selection:

SVC and Logistic Regression presented an accuracy higher than 90%, with fast training and efficient predictions. We saved the trained models into pickle files to build an app.

StreamLit App:

A built in the platform StreamLit, where model SVC was loaded from pickle file. User interface allows user to give a prompt and the app returns a value of 'positive', 'negative' or 'neutral'

Conclusions

Conclusion 1:

Due to the standardization process of transforming ratings to 3 categories, the model is not very efficient classifying 'neutral' reviews.

Conclusion 2:

The model has a high accuracy when differentiating between 'positive' and 'negative' reviews

Conclusion 3:

Data cleaning was critical for our model to properly classify reviews.