Best Model for Sentiment Analysis on Amazon Reviews

By Dominique Brown

Introduction:

A company's success is dependent on their ability to receive feedback and adapt to the customers needs. One way for a company to improve or determine viability of a product is to receive reviews from the customers after they have bought the product or service. For companies like Amazon, there can be thousands of reviews for each product so digesting that information can be too complicated manually.. One way to get a general sense of the sentiment of the reviews of the product is to do a sentiment analysis. This experiment determines the best model and parameters for a set of amazon reviews.

Background:

The analysis being used for this experiment is sentiment analysis. A sentiment analysis is a text analysis method to interpret emotions within the text. Sentiment analysis is used by businesses to understand how customers feel about a product. A sentiment analysis classifies the text into positive, neutral, and negative. One way to prepare the data to train the model is to use a CountVectorizer. A CountVectorizer extracts the data to count the number of words (term frequency), limit your vocabulary size, apply stop words and etc (Pedregosa).” Another tool that can be used is a TfidfTransformer which can then read what was extracted with CountVectorizer into a readable form for the model to analyze the text for sentiment words. In this experiment, three models are being tested for accuracy. MultinomialNB model is based on Naive Bayes. Naive Bayes is a statistical model to classify and uses the Bayes Theorem. One benefit of the Bayes Theorem is that it is highly scalable (Gandhi). LogisticRegression is another model that is used in this experiment. Logistic regression is a statistical model that predicts discrete values and is used for classification (Swaminathan). LinearSVC (Support Vector Classifier) is the last model used in this experiment. A Linear SVC model job is “to fit to the data you provide, returning a "best fit" hyperplane that divides, or categorizes, your data (Python Programming).” LinearSVC is a SVM (Support Vector Machine). SVM are supervised learning methods that are used for classification and regression.

The data used in this sentiment analysis are reviews from cell phones and accessories that are sold on amazon. The data includes a rating (1-5), verified buyer, reviewer ID, reviewer name, ReviewText (body of review), summary (title of review), review time (date of review), image, and vote.

Methodology:

To answer the question, “Which classification/ regression model will be the most efficient to perform a sentiment analysis on amazon reviews?”, two main parts were used in performing this task, preparing the data and training the models.

First, the data needed to be prepared. The data came from the website http://deepyeti.ucsd.edu/jianmo/amazon/categoryFilesSmall/Cell_Phones_and_Accessories_5.json.gz. The data was collected into Pandas lines of code that brought the data files from the website. Once the data was collected into Pandas, the data was read by read_json() command with the parameter of reviews.json and line=True. Once the data was read, exploration was done using the command df.head() and df.shape to see the columns and what the data looked like. The columns that were needed to do the analysis were ReviewText, summary, and overall. ReviewText and summary had the words that were to be analyzed and overall was the rating that would be used in the sentiment target analysis. Based on looking at the data, some columns that could be dropped are vote, image, and style because they had many missing values and they are not useful for the analysis. The code, df1=df.drop(columns=['vote', 'image','style'], was used to drop the columns. For each transformation, the data frame is renamed to avoid confusion. The next step in preparing the data for analysis was to streamline the analysis of the words in the review. To do this, ReviewText and summary columns are concatenated together to create one column. The code, df1["review"] = df1["reviewText"] + df1["summary"], is used to concatenate the columns together and the reviewText and summary are dropped from the dataframe using df2=df1.drop(columns=['reviewText', 'summary']). The next step is to prepare the data to be in a training set and a test set. To do this, StratifiedShuffleSplit was used to split the data randomly and equally into a training set and a test set. The sentiments were created with >3 being positive, 3 being neutral, and <3 being negative. The sentiments were then applied to the overall column to each set. This created the target that the models would run an analysis on. The final step to prepare the data is the use of Count Vectorizer and a TfidfTransformer. The Count Vectorizer converts text into a matrix of tolkien counts. A TfidfTransformer transforms a count matrix from text into a more streamlined reading process for the model. Using both Count Vectorizer and TfidfTransformer made all models more accurate than just using one.

The last part of the experiment, now that the data is prepared is to perform the models. MultinomialNB, LogisticRegression, and LinearSVC are used to train the data to classify the sentiments based on the text reviews. A pipeline is used to include Count Vectorizer and TfidfTransformer in the model. MultinomialNB was chosen to try because it is most often used for text analysis. Logistic Regression is good for classifying so it was chosen to try out as well. LinearSVC performs classifying and it scales better for large data sets. Accuracy is then calculated for each model to determine which model to go with. The command accuracy_score() was used because it works for all models and can be compared across models. A Grid Search is then used to determine the specific parameters for the model with the best accuracy.

Results:

To decide on a model that is the best estimator for sentiment analysis on this data of amazon reviews, an accuracy score is used to determine the best estimator model. Three models were tested for the best accuracy in estimating the sentiment analysis of the amazon reviews. The first model, MultinomialNB (Naive Bayes), had an accuracy of 0.8296231966254298. The second model, logistic regression, had an accuracy of 0.8953821204494701. The final model, LinearSVC (SVM), had an accuracy of 0.8988825280918791. Out of three models, the LinearSVC had the best accuracy. A grid search was then done on the MultinomialNB model with the parameters vect__ngram_range and tfidf__use_idf. The grid search brought the accuracy up to 0.9146786714402183. The best parameter to get to the best accuracy is 'tfidf__use_idf': True and 'vect__ngram_range': (1, 2).

Conclusion:

After performing three separate models, the experiment led to the conclusion that the LinearSVC had the highest accuracy. When a grid search is performed, best accuracy is with the parameters 'tfidf__use_idf': True and 'vect__ngram_range': (1, 2). Another conclusion this experiment led too was how to evaluate multiple columns in a model. The path this experiment took was to concatenate the columns. This experiment can help a company who sells products on Amazon understand the overall sentiment of their product.

Works Cited:

Gandhi, R. (2018, May 17). Naive Bayes Classifier. Retrieved from https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

Pedregosa *et al.* (2011). Scikit-learn: Machine Learning in Python, , JMLR 12, pp. 2825-2830

Python Programing. Linear SVC Machine learning SVM example with Python. (n.d.). Retrieved from https://pythonprogramming.net/linear-svc-example-scikit-learn-svm-python/

Swaminathan, S. (2019, January 18). Logistic Regression - Detailed Overview. Retrieved from https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc

Zhang, Mick. (n.d.). Amazon Reviews using Sentiment Analysis. Retrieved from https://github.com/mick-zhang/Amazon-Reviews-using-Sentiment-Analysis/blob/master/Amazon Project Github.ipynb