NLP Analysis Of Restaurant Reviews Project Report

Protim Mondal

Considering how difficult it is to launch a business, like a restaurant. For a restaurant owner, reading all of the assessments and comparing input to other establishments becomes difficult and time consuming. As a result of this project, it is an important factor to consider while choosing a restaurant. This study presents an efficient restaurant review prediction algorithm for predicting restaurant evaluations based on a specific set of client feedback. Restaurant reviews are labeled with a 0 or a 1, with 0 indicating a bad review and 1 indicating a positive one. The study's findings could be used as a long-term marketing strategy for review website owners, allowing customers to sort and filter valuable evaluations depending on their preferences.

Restaurant reviews provide textual information. **ML** enables systems to learn without being explicitly programmed, and this learning can be utilized to solve problems. Machine learning provides **Natural Language Processing (NLP)** skills for text processing. We may easily analyze our textual datasets using **NLP** methods. Data analysts can use **NLP** to apply machine learning and deep learning algorithms to their textual datasets. For classifying reviews, we employ machine learning algorithms.

Methodology:

The dataset is a .tsv file containing 1000 text-based reviews retrieved from www.kaggle.com. This dataset is used to train the **Random Forest classifier** and **SVM model.** It's split into two sections: 75% training data and 25% test data. The dataset contains two columns. The first column includes written reviews from various people associated with the restaurant's food as well as an overall evaluation of the facility. The second column displays the mood, or whether the review is positive or unfavourable. The number 1 represents a positive review, while the number 0 represents a negative review. Transform the dataset to a Pandas Data frame after importing it. The model should be able to predict whether the review will be favourable or unfavourable. After the dataset has been cleansed from 1000 reviews and the reviews that are not suitable have been eliminated from the dataset, the Data Frame is delivered as input to the **Countvectorizer** for further processing.

Cleaning the texts:

It is not possible to go directly from raw text to fitting a machine learning model. We must first clean our material, which includes breaking it down into words and dealing with punctuation and case. In fact, we may need to utilize a variety of text preparation strategies, and the way we choose is entirely dependent on our **Natural Language Processing** requirement.

After obtaining our text data, the first step in cleaning up text data is to have a clear notion of what we want to achieve and then evaluate our text to determine what specifically might help.

Here’s the features of the dataset:

- It's plain text so there is no markup to parse (yay!).
- The translation of the original German uses UK English (e.g. "travelling").
- The lines are artificially wrapped with new lines at about 70 characters (meh).
- There are no obvious spelling mistakes.
- There's punctuation like commas, apostrophes, quotes, question marks, and more.
- There's hyphenated descriptions like "armour-like".
- There's a lot of use of them dash ("-") to continue sentences (maybe replace with commas?).
- There are names (e.g. "Mr. Samsa")
- There does not appear to be numbers that require handling (e.g. 1999)
- There are section markers (e.g., "II" and "III"), and we have removed the first "I".

Tokenization:

Tokenization is done by natural language toolkit predefined function **nltk.word_tokenize**. **NLTK** is a platform for python language, it offers over 50 corpuses and lexical resources for example sentiwordnet3.0, Wordnet etc. NLTK provides suite to process textual data, like tokenization, tagging, semantic reasoning, stemming for strengthen **NLP libraries.** **Converting into lower case:**

The main reason for converting text to lower case is that when we are going to remove stop words from the actual data, the reviews may have multiple uppercase and lowercase combinations. There are only two ways to remove all stop words from the data. The first is to identify each and every word that is present in the data and then remove from the data, which will be a difficult and timeconsuming effort. Another option is to use the **NLTK corpus** of stop words and convert them to comparable case and same can be eliminated from the data.

Removing Stop Words:

According to standford.edu, some particularly common words, those word's appearance have little value in favouring select documents matching, a user need are excluded from the vocabulary entirely. Those words are stop words.

Creating the Bag of Words model:

A metric for the presence of well-known terms. Because any information about the sequence or structure of words in the document is deleted, it is referred to as a "bag" of words. The model doesn't care where in the document a known word appears; it merely cares about whether it does. We take all the different words of reviews in the dataset without repeating of words. One column for each word. If a word is there in the row of the dataset of reviews, then the count of the word will be there in the row of a bag of words under the column of the word.

We'll need the **CountVectorizer** class from **sklearn.feature_extraction.text** for this. We may also specify a maximum number of features (through the variable **"max_features"**) that will be used. Perform the training on the corpus, then apply the same transformation to it **".fit_transform(corpus)"** and convert it to an array. If the review is positive or negative, the answer is in the dataset's second column[:, 1]: all rows and first column (indexing from zero).

Splitting the dataset into the Training set and Test set:

The train-test split procedure is used to estimate the performance of machine learning algorithms when they are used to make predictions on data not used to train the model. Here we are splitting **Corpus** into Training and Test set. For this, we need class **train_test_split** from **sklearn.model_selection.**

Here X : Bag of words.

Y : 0 or 1 (positive and negative).

<u>Training the Random Forest classifier model on the Training set:</u>

Since **Random forest** is an ensemble model (made of many trees) from **sklearn. ensemble,** import **RandomForestClassifier** class with 501 trees or **"n_estimators"** and criterion as 'entropy' Fit the model via **.fit()** method with attributes X_train and y_train. One of the main advantages of **random forest** for **NLP** is their ability to handle high-dimensional and sparse datasets, which are common in **text analysis**. Random forests can also deal with missing values, outliers, and imbalanced classes, which can affect the performance of other algorithms. Moreover, random forests are easy to implement and tune, as they have few hyperparameters and do not require much preprocessing or scaling of the data.

<u>Training the SVM model on the Training set:</u>

The study developed a Machine Learning model that could help with **restaurant review categorisation.** This model was trained using the **SVM (Support Vector Machine)** Algorithm to categorize the reviews. **SVM** is an abbreviation for Supervised Learning, and it is used to tackle classification and regression problems. However, it is mostly used in Machine Learning for classification problems. SVMs may perform non-linear and linear classification efficiently by utilizing a technique or parameter known as **Kernel,** which implicitly transforms their inputs into high-dimensional feature spaces. From **sklearn.svm,** import **SVC** with **kernel='linear'** and **random_state=50** Fit the model via **.fit()** method with attributes X_train and y_train. is one of the most commonly used clustering algorithms in industrial applications.

<u>CONCLUSION:</u>

Based on 1000 customer evaluations, this study proposes an efficient restaurant review prediction system for predicting the review for the restaurant. The numbers 0 and 1 are used to represent restaurant reviews, with 0 representing a negative review and 1 denoting a positive one. **Natural Language Processing (NLP)** in conjunction with the **Random Forest** classification algorithm produced the prediction accuracy 72.4% (It may be different when performed an experiment with different test sizes, here = 0.25).

And on the other hand, **Natural Language Processing (NLP)** combined with the **SVM classification** method yielded the greatest prediction accuracy-(80%).

This method will assist business owners in predicting and improving client feedback. As a result of this research, it is an important consideration to consider while choosing a restaurant. It is an essential part of the planning process when launching a business, such as a restaurant.

<u>FUTURE WORK:</u>

Now all the works are related to binary sentiment classification i.e., positive or negative, in future we can extend it to degree of positivity and negativity with reviews. Restaurant reviews often contain not only text but also visual elements, such as photos and videos. Future work could extend on how to integrate these multimodal features into sentiment analysis to improve the accuracy and depth of analysis.