NATURAL LANGUAGE PROCESSING

# RESTAURANT REVIEW SYSTEM

## USING NAIVE BAYES AND LOGISTIC REGRESSION IN PYTHON

**BY : JIVESH PODDAR (16BIT020)**

## INTRODUCTION

Many online web portals allow customers to share their experience and rate a given restaurant based on it. The amount of reviews generated is huge and it is an enormous task to analyze them and generate some useful information that a new customer can utilize.Over the last decade, we have moved from scarcity of information to an abundance of information. Due to the plethora of restaurant choices today, restaurant selection has become really difficult. Although it allows us to read reviews and understand the qualities of a restaurant ourselves, it is really hard to get a holistic view of the restaurant without reading all the reviews.Consider an alternate scenario in which you can read a summary of any restaurant with its pros and cons and a rating for each of the categories Food, Ambience and Service. You can keep these short summaries side by side and efficiently compare two restaurants in a matter of seconds ! In this project, we aim to generate significant attributes of a restaurant so that it gets ease for customers to choose whether to visit that restaurant or not by classifying the descriptors into pros and cons, thus generating a visually appealing and quickly readable summary of the restaurant.

## Literature survey

In previous implementations, different approaches have taken place in order to analyze restaurant reviews by customers by using Natural Language Processing and Machine Learning techniques. , which summarizes user descriptions for various dishes by means of LDA topic modelling, another system finds sentiments related to various aspects of a restaurant such as the food, service, price, ambience, by making use of the opinionated words used in each review sentence. Another built a clustering based system which clusters restaurant with similar food genres and finds unique features in each restaurant genre. These researches are reviewed in detail in later section.

Since this research focuses on improving the customer experience at restaurants by analysing text reviews, previous studies from the field of text analytics, Natural Language Processing and Machine learning algorithms are studied which are prominently based in hospitality industry and improving customer experience.

- **Tokenization** : It is the process of breaking up the given text into units called tokens. The tokens may be words or number or punctuation mark. Tokenization does this task by locating word boundaries. Ending point of a word and beginning of the next word is called word boundaries. Tokenization is also known as word segmentation. Tokenizing means splitting your text into minimal meaningful units. It is a mandatory step before any kind of processing.The basic tokenizer (like in NLTK) will split your text into sentences and your sentences into typographic tokens.

- **Stemming :** Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. Stemming is important in natural language understanding (NLU) and natural language processing (NLP). Stemming is a part of linguistic studies in morphology and artificial intelligence (AI) information retrieval and extraction. Stemming and AI knowledge extract meaningful information from vast sources like big data or the Internet since additional forms of a word related to a subject may need to be searched to get the best results. Stemming is also a part of queries and Internet search engines.

- **Lemmatization :** Lemmatization is the process of converting a word to its base form. The difference between stemming and lemmatization is, lemmatization considers the context and converts the word to its meaningful base form, whereas stemming just removes the last few characters, often leading to incorrect meanings and spelling errors. Lemmatization in linguistics is the process of grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's lemma, or dictionary form.In computational linguistics, lemmatization is the algorithmic process of determining the lemma of a word based on its intended meaning. Unlike stemming, lemmatization depends on correctly identifying the intended part of speech and meaning of a word in a sentence, as well as within the larger context surrounding that sentence, such as neighboring sentences or even an entire document. As a result, developing efficient lemmatization algorithms is an open area of research.

- **POS Tagging :** It is the process of marking up a word in a corpus to a corresponding part of a speech tag, based on its context and definition. This task is not straightforward, as a particular word may have a different part of speech based on the context in which the word is used. For example: In the sentence "Give me your answer", *answer* is a Noun, but in the sentence "Answer the question", *answer* is a verb.To understand the meaning of any sentence or to extract relationships and build a knowledge graph, POS Tagging is a very important step.

## Dataset Description

Many popular datasets from the restaurant domain are used in previous research which primarily focuses on sentiment analysis. A few of the datasets available are already annotated with some useful metadata such as sentiment score, sentiment polarity, aspects contained in the reviews, which promotes the researchers with some extra information to work on.

A simple project in python which reads a tsv file, cleans the restaurant reviews(text), generates a bag-of-words model and uses a classifier which tells whether the review is a positive one or a negative one

*Description of the dataset to be used:*

- *Columns separated by \t (tab space)*
- *First column is about reviews of people*
- *In second column, 0 is for negative review and 1 is for positive review*

*# Importing the Libraries*

*Import numpy as np*

*Import matplotlib.pyplot as plt*

*Import pandas as pd*

*# Importing the dataset*

*dataset = pd.read_csv('c:\\Restaurant_Reviews.tsv', delimiter = '\t', quoting = 3)*

## Proposed Method

### Data Pre-Processing

**#Tokenization**

```
from nltk.tokenize import sent_tokenize, word_tokenize
```

**# Cleaning the texts by Removing Stopwords and showing Frequency Distribution.**

```
import re

import nltk

from nltk.corpus import stopwords

from nltk import FreqDist
```

**#STEMMING**

```
from nltk.stem.porter import PorterStemmer
```

**#LEMMATIZATION: LEXICON NORMALIZATION**

```
from nltk.stem.wordnet import WordNetLemmatizer
```

**#POS TAGGING**

### Implementation

- Firstly the data is preprocessed , i.e. reviews are combined put in same format and punctuation marks are removed.
- Then based upon the techniques of POS tagging , Stemming and Lemmatization we extract the meaningful words out of the sentence.
- Then we find out the Part of Speech used.
- Then the Data is Trained through the 2 approaches after Fitting.
- 10% of the training reviews are separated for testing.
- Test Results are Obtained after training and Confusion Matrix is calculated.
- Accuracy , Precision and Recall is also calculated.

*# Creating the Bag of Words model*

*from sklearn.feature_extraction.text import CountVectorizer*

*cv = CountVectorizer(max_features = 1500)*

*result=arrayconversion(corpus)*

*print(result)*

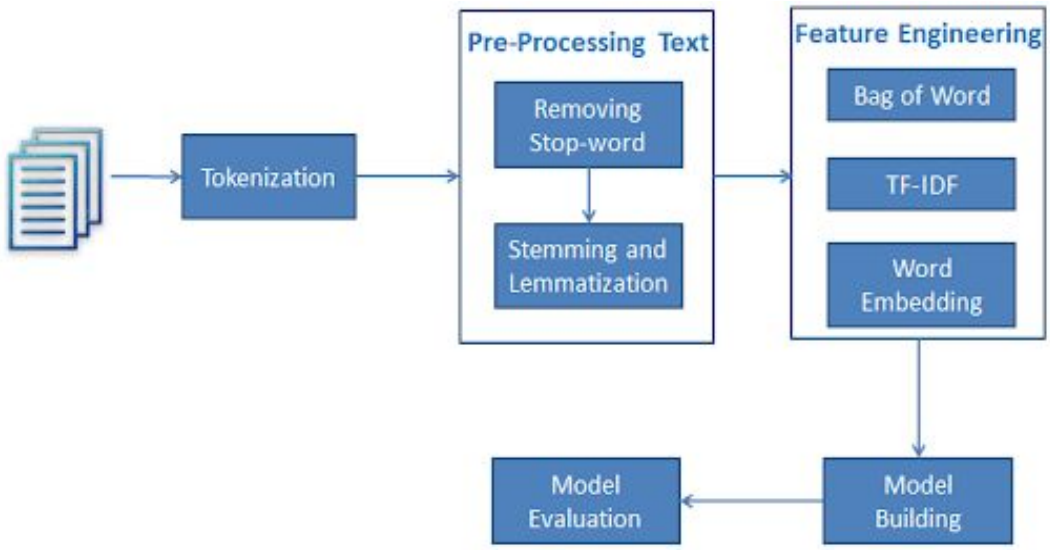
*X = cv.fit_transform(result)*

*y = dataset.iloc[:, 1].values*

*# Splitting the dataset into the Training set and Test set*

*from sklearn.model_selection import train_test_split*

*X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.10, random_state = 0)*



## Classification Methods Used

### Naive Bayesian Model

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. Naive Bayes model is easy to build and particularly useful for very large datasets. Along with simplicity, Naive Bayes is known to outperform even the most-sophisticated classification methods. It proves to be quite robust to irrelevant features, which it ignores. It learns and predicts very fast and it does not require lots of storage. So, why is it then called naive? The naive was added to the account for one assumption that is required for Bayes to work optimally: all features must be independent of each other. In reality, this is usually not the case; however, it still returns very good accuracy in practice even when the independent assumption does not hold.

```
# Fitting Naive Bayes to the Training set

from sklearn.naive_bayes import GaussianNB

classifier = GaussianNB()

classifier.fit(X_train, y_train)
```

### Logistic Regression Model

Logistic regression is generally used where the dependent variable is Binary or Dichotomous. That means the dependent variable can take only two possible values such as "Yes or No", "Default or No Default", "Living or Dead", "Responder or Non Responder", "Yes or No" etc. Independent factors or variables can be categorical or numerical variables.

Please note that even though logistic (logit) regression is frequently used for binary variables (2 classes), it can be used for categorical dependent variables with more than 2 classes. In this case it's called Multinomial Logistic Regression.

Here we will focus on Logistic Regression with binary dependent variables as it is most commonly used.

*# Fitting Logistic Regression to the Training set*

*from sklearn import linear_model*

*classifier = linear_model.LogisticRegression(C=1.5)*

*classifier.fit(X_train, y_train)*

## Results

*AFTER POS TAGGING*

*Liked Wow... Loved this place. 1*

*Total words left is : 3 out of 6*

*Lemmatized words are : wow*

*Lemmatized words are : love*

*Lemmatized words are : place*

*[('wow', 'NNS'), ('love', 'VBP'), ('place', 'NN')]*

## Results using Naive Bayes Model

*# Predicting the Test set results*

*y_pred = classifier.predict(X_test)*

*print(y_pred)*

*[ 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 0 0 1 1 1 0 1 1 1 0 1 1 1 1 1 1 0 1  0 1 1 1 1 1 0 1 0 1 1 0 0 1 1 1 1 0 0 1 1 0  1 1 0 1 1 1 0 1 1 1 1 1 1 1 1  0 1 1 0 0 1 0 1 1 0 1 1 0 0 1 1 0 1 0 0 1 1 1 1 1 1 ]*

*# Making the Confusion Matrix*

*from sklearn.metrics import confusion_matrix*

*cm = confusion_matrix(y_test, y_pred)*

*print(cm)*

*Confusion Matrix:*

*[[ 27 24]*

*[ 3 46 ]]*

*Accuracy is  73.00 %*

*Precision is  0.53*

*Recall is  0.37*

## Results using Logistics Regression Model

*# Predicting the Test set results*

*y_pred = classifier.predict(X_test)*

*print(y_pred)*

*[ 0 0 0 0 0 0 1 0 0 1 1 1 1 1 1 1 0 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 1 1 1 0 1 1 1 1 1 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 1 1 0 0 0 0 0 1 0 1 1 1 1 0 1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1 ]*

*# Making the Confusion Matrix*

*from sklearn.metrics import confusion_matrix*

*cm = confusion_matrix(y_test, y_pred)*

*print ("Confusion Matrix:\n",cm)*

*Confusion Matrix:*

*[[38 13]*

*[13 36]]*

*Accuracy is 74.00 %*

*Precision is 0.745*

*Recall is 0.51*

## Conclusion & Future Work

This research presented a novel approach on distinguishing the food dishes served at a restaurant by classifying the reviews based on the previous customer reviews.

”Using Machine Learning and Natural Language Processing techniques, we can distinguish the best dishes served at a restaurant from the bad ones”

”Using Machine Learning and Natural Language Processing techniques, we can choose whether to visit that restaurant or not”

Overall genre of the restaurant can be predicted.

This research is primarily focused on analysing a restaurant reviews, but by Increasing the scope of the restaurants , to some other domain specific dictionary, a similar approach can be applied to future studies as well.

## References

http://spacab.com/wp/using-python-to-perform-lexical-analysis-on-a-short-story

https://www.datacamp.com/community/tutorials/text-analytics-beginners-nltk

https://www.nltk.org/book/ch08.html

http://trap.ncirl.ie/3426/1/kedarratnaparkhi.pdf

https://pdfs.semanticscholar.org/da73/63c004cd28f8f3c423cc9a0a286d492eb904.pdf