

**HMR INSTITUTE OF TECHNOLOGY & MANAGEMENT**

**HAMIDPUR , DELHI - 110036**

**Affiliated to**

**GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY**

**Sector - 16C Dwarka, Delhi - 110075, India 2008-12**

# WINTER TRAINING PROJECT On RESTAURANT REVIEWS CLASSIFICATION

Submitted in partial fulfilment of the requirements for

The award of the degree of

**Bachelor of Technology  
In**

**(ELECTRONICS AND COMMUNICATION ENGINEERING)**

**Guide(s): Submitted by:**

**MR. RAHUL pathak YASH CHAUHAN**

**Roll No:01613302817**

DECLARATION

We, students of B.Tech (ECE 6th sem) hereby declare that the minor project entitled “RESTAURANT REVIEWS CLASSSIFICATION ” which is submitted to the Department of ECE, HMR Institute of Technology & Management, Hamidpur Delhi, affiliated to Guru Gobind Singh Indraprastha University, Dwarka(New Delhi) in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in ECE, has not been previously formed the basis for the award of any degree, diploma or other similar title or recognition. The list of member(s)involved in the project is listed below: -

* **YASH CHAUHAN (01613302817)**

This is to certify that the above statement made by the candidate(s) is correct to the best of my knowledge.

**Mr. A.K. Sharma**

**(Head of Department)**

**(Assistant Professor of Electronics Department)**

**HMRITM Hamidpur, New Delhi-110036**

CERTIFICATE

This is to certify that the project entitled **“**RESTAURANT REVIEWS CLASSSIFICATION**”** is the bonafide work carried out by YASH CHAUHANstudent of **B.Tech , HMR Institute of Technology & Management, Hamidpur Delhi, affiliated to Guru Gobind Singh Indraprastha University, Dwarka(New Delhi) ,** during the year **3RD YEAR,** in partial fulfillment of the requirements for the award of the Degree of Bachelor of **Electronic and Communication Engineering** and that the project has not formed the basis for the award previously of any degree, diploma, associateship, fellowship or any other similar title.

**Signature of the Guide : RAHUL PATHAK**

**Place: HMRITM**

**Date: 01/02/2020**

ACKNOWLEDGEMENT

**It is my pleasure to be indebted to various people, who directly or indirectly contributed in the development of this work and who influenced my thinking, behavior, and acts during the course of study.**

**I express my sincere gratitude to V C PANDEY, worthy Principal for providing me an opportunity to undergo winter training at HMR Institute of Technology**

**I am thankful to Mr. Rahul for his support, co-operation, and motivation provided to me during the training for constant inspiration, presence and blessings.**

**I also extend my sincere appreciation to Mr/Ms Rahul who provided his valuable suggestions and precious time in accomplishing my project report.**

**Lastly, I would like to thank the almighty and my parents for their moral support and my friends with whom I shared my day-to-day experience and received lots of suggestions that improved my quality of work.**

**Name of the student:**

* yash Chauhan
* ajay mittal
* satyam kumar

Contents

## 1. Cover Page & Title page

## 2. Declaration by the Student

## 3. Certificate by the Guide

## 4. Acknowledgment

## 5.Team details

## 6. Abstract

## 7. Table of Contents

## 8. Codes

## 9. Appendices

## 10. References

Team Details

Yash chauhan

Roll No:01613302817

Contribution:report file

Satyam kumar dubey

Roll No: 01313302817

Contribution: Coding

Ajay mittal

Roll No: 42113302817

Contribution: coding

Abstract

This project aims at “RESTAURANT REVIEWS CLASSIFICATION” according to their choices of teams. In this project we are analysing large amount of natural language data in other words we can say that we are extracting meaningfull information from textfull Data.

General steps of Natural Processing Are:-

* Remove all the punctuations, numbers, emojis, symbols from the Dataset.
* Get all the Data into similar cases.
* Remove all the unwanted words like Preposition, Conjuction, etc.
* Perform stemming or lemmatization.
* Use a ML and Deep Learning model to describe the Data.

NATURAL LANGUAGE PROCESSING(NLP):-

It is a field of computer science, artificial intelligence and computational linguistics concerned with the interactions between computers and human languages, and, in particular, concerned with programming computers to fruitfully process large natural language corpora. The Python packages that we use in this notebook are: numpy, pandas, matplotlib, re, nltk .

Since usually such tutorials are based on in-built datasets like iris, It becomes harder for the learner to connect with the analysis and hence learning becomes difficult. To overcome this, The dataset that we use in this notebook is amazon review dataset posted on Kaggle.com . AMAZON is one of the most popular e-commerce website for purchase products online in the world, thus the problems we try to solve and the questions that we try to answer should be familiar to anyone who knows about AMAZON.

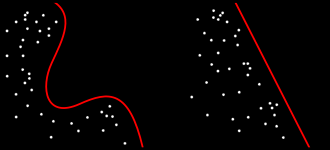
Algorithm Used

naive\_bayes:

In [machine learning](https://en.wikipedia.org/wiki/Machine_learning), **naïve Bayes classifiers** are a family of simple "[probabilistic classifiers](https://en.wikipedia.org/wiki/Probabilistic_classification)" based on applying [Bayes' theorem](https://en.wikipedia.org/wiki/Bayes%27_theorem) with strong (naïve) [independence](https://en.wikipedia.org/wiki/Statistical_independence) assumptions between the features. They are among the simplest [Bayesian network](https://en.wikipedia.org/wiki/Bayesian_network) models.[[1]](https://en.wikipedia.org/wiki/Naive_Bayes_classifier#cite_note-1)

Naïve Bayes has been studied extensively since the 1960s. It was introduced (though not under that name) into the [text retrieval](https://en.wikipedia.org/wiki/Information_retrieval) community in the early 1960s,[[2]](https://en.wikipedia.org/wiki/Naive_Bayes_classifier#cite_note-2) and remains a popular (baseline) method for [text categorization](https://en.wikipedia.org/wiki/Text_categorization), the problem of judging documents as belonging to one category or the other (such as [spam or legitimate](https://en.wikipedia.org/wiki/Spam_filtering), sports or politics, etc.) with [word frequencies](https://en.wikipedia.org/wiki/Bag_of_words) as the features. With appropriate pre-processing, it is competitive in this domain with more advanced methods including [support vector machines](https://en.wikipedia.org/wiki/Support_vector_machine).[[3]](https://en.wikipedia.org/wiki/Naive_Bayes_classifier#cite_note-rennie-3) It also finds application in automatic [medical diagnosis](https://en.wikipedia.org/wiki/Medical_diagnosis).[[4]](https://en.wikipedia.org/wiki/Naive_Bayes_classifier#cite_note-rish-4)

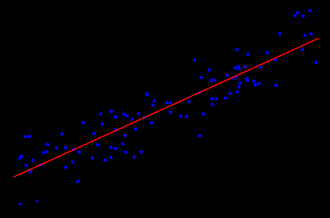
Naïve Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. [Maximum-likelihood](https://en.wikipedia.org/wiki/Maximum-likelihood_estimation) training can be done by evaluating a [closed-form expression](https://en.wikipedia.org/wiki/Closed-form_expression),[[5]](https://en.wikipedia.org/wiki/Naive_Bayes_classifier#cite_note-aima-5):718 which takes [linear time](https://en.wikipedia.org/wiki/Linear_time), rather than by expensive [iterative approximation](https://en.wikipedia.org/wiki/Iterative_method) as used for many other types of classifiers.



Logistic Regression:

In [statistics](https://en.wikipedia.org/wiki/Statistics), the **logistic model** (or **logit model**) is used to model the probability of a certain class or event existing such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events such as determining whether an image contains a cat, dog, lion, etc. Each object being detected in the image would be assigned a probability between 0 and 1 and the sum adding to one.

Logistic regression is a [statistical model](https://en.wikipedia.org/wiki/Statistical_model) that in its basic form uses a [logistic function](https://en.wikipedia.org/wiki/Logistic_function) to model a [binary](https://en.wikipedia.org/wiki/Binary_variable) [dependent variable](https://en.wikipedia.org/wiki/Dependent_variable), although many more complex [extensions](https://en.wikipedia.org/wiki/Logistic_regression#Extensions) exist. In [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis), **logistic regression**[[1]](https://en.wikipedia.org/wiki/Logistic_regression#cite_note-1) (or **logit regression**) is [estimating](https://en.wikipedia.org/wiki/Estimation_theory) the parameters of a logistic model (a form of [binary regression](https://en.wikipedia.org/wiki/Binary_regression)). Mathematically, a binary logistic model has a dependent variable with two possible values, such as pass/fail which is represented by an [indicator variable](https://en.wikipedia.org/wiki/Indicator_variable), where the two values are labeled "0" and "1". In the logistic model, the [log-odds](https://en.wikipedia.org/wiki/Log-odds) (the [logarithm](https://en.wikipedia.org/wiki/Logarithm) of the [odds](https://en.wikipedia.org/wiki/Odds)) for the value labeled "1" is a [linear combination](https://en.wikipedia.org/wiki/Linear_function_(calculus)) of one or more [independent variables](https://en.wikipedia.org/wiki/Independent_variable) ("predictors"); the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a [continuous variable](https://en.wikipedia.org/wiki/Continuous_variable) (any real value). The corresponding [probability](https://en.wikipedia.org/wiki/Probability) of the value labeled "1" can vary between 0 (certainly the value "0") and 1 (certainly the value "1"), hence the labeling; the function that converts log-odds to probability is the logistic function, hence the name. The [unit of measurement](https://en.wikipedia.org/wiki/Unit_of_measurement) for the log-odds scale is called a [*logit*](https://en.wikipedia.org/wiki/Logit), from ***log****istic un****it***, hence the alternative names.



Linear Regression:

In statistics, linear regression is a linear approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one independent variable, the process is called multiple linear regression. This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.

In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models. Most commonly, the conditional mean of the response given the values of the explanatory variables (or predictors) is assumed to be an affine function of those values; less commonly, the conditional median or some other quantile is used. Like all forms of regression analysis, linear regression focuses on the conditional probability distribution of the response given the values of the predictors, rather than on the joint probability distribution of all of these variables, which is the domain of multivariate analysis.

Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications.This is because models which depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters and because the statistical properties of the resulting estimators are easier to determine.

3.2. Assumptions

Standard linear regression models with standard estimation techniques make a number of assumptions about the predictor variables, the response variables and their relationship. Numerous extensions have been developed that allow each of these assumptions to be relaxed (i.e. reduced to a weaker form), and in some cases eliminated entirely. Generally these extensions make the estimation procedure more complex and time-consuming, and may also require more data in order to produce an equally precise model.

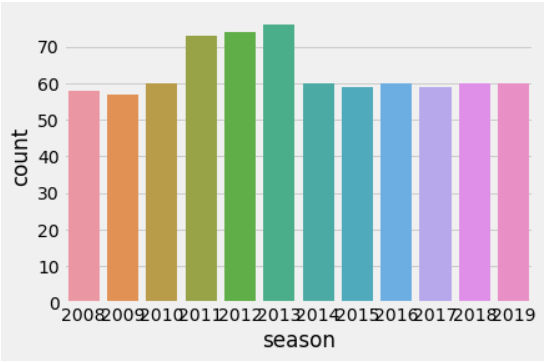


Fig. 3.2 (Figure Description)

Example of a cubic polynomial regression, which is a type of linear regression.

Libraries Used

Numpy:

NumPy is a library for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)), adding support for large, multi-dimensional [arrays](https://en.wikipedia.org/wiki/Array_data_structure) and [matrices](https://en.wikipedia.org/wiki/Matrix_(math)) , along with a large collection of [ematical](https://en.wikipedia.org/wiki/High-level_programming_language) [functions](https://en.wikipedia.org/wiki/Function_(mathematics)) to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by [Jim Hugunin](https://en.wikipedia.org/wiki/Jim_Hugunin) with contributions from several other developers. In 2005, [Travis Oliphant](https://en.wikipedia.org/wiki/Travis_Oliphant) created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is [open-source software](https://en.wikipedia.org/wiki/Open-source_software) and has many contributors.

* History

The Python programming language was not initially designed for numerical computing, but attracted the attention of the scientific and engineering community early on, so that a special interest group called matrix-sig was founded in 1995 with the aim of defining an array computing package. Among its members was Python designer and maintainer, who implemented extensions to [Python's syntax](https://en.wikipedia.org/wiki/Python_syntax_and_semantics) (in particular the indexing syntax) to make array computing easier.There was a desire to get Numeric into the Python standard library, but Guido van Rossum decided that the code was not maintainable in its state then. In early 2005, NumPy developer [Travis Oliphant](https://en.wikipedia.org/wiki/Travis_Oliphant) wanted to unify the community around a single array package and ported Numarray's features to Numeric, releasing the result as NumPy 1.0 in 2006.[[6]](https://en.wikipedia.org/wiki/NumPy) This new project was part of SciPy. To avoid installing the large SciPy package just to get an array object, this new package was separated and called NumPy. Support for Python 3 was added in 2011 with NumPy version 1.5.0.In 2011, PyPy started development on an implementation of the NumPy API for PyPy. It is not yet fully compatible with NumPy.

Pandas:

In [computer programming](https://en.wikipedia.org/wiki/Computer_programming), pandas is a [software library](https://en.wikipedia.org/wiki/Software_library) written for the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)) for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and [time series](https://en.wikipedia.org/wiki/Time_series). It is [free software](https://en.wikipedia.org/wiki/Free_software) released under the [three-clause BSD license](https://en.wikipedia.org/wiki/3-clause_BSD_license). The name is derived from the term "[panel data](https://en.wikipedia.org/wiki/Panel_data)", an [econometrics](https://en.wikipedia.org/wiki/Econometrics) term for data sets that include observations over multiple time periods for the same individuals.

**Pandas** is the most popular **python** library that is used for data analysis. It provides highly optimized performance with back-end source code is purely written in C or **Python**. We can analyze data in **pandas** with: Series

Matplotlib:

Matplotlib. Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+.There is also a [procedural](https://en.wikipedia.org/wiki/Procedural_programming) "pylab" interface based on a [state machine](https://en.wikipedia.org/wiki/State_machine) (like [OpenGL](https://en.wikipedia.org/wiki/OpenGL)), designed to closely resemble that of [MATLAB](https://en.wikipedia.org/wiki/MATLAB), though its use is discouraged.[[3]](https://en.wikipedia.org/wiki/Matplotlib#cite_note-3) [SciPy](https://en.wikipedia.org/wiki/SciPy) makes use of Matplotlib.

Matplotlib was originally written by [John D. Hunter](https://en.wikipedia.org/wiki/John_D._Hunter), has an active development community, and is distributed under a [BSD-style license](https://en.wikipedia.org/wiki/BSD_licenses). Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012, and further joined by Thomas Caswell.

Seaborn:

Statistical data visualization. Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Seaborn is a Python data visualization library based on [matplotlib](https://matplotlib.org/). It provides a high-level interface for drawing attractive and informative statistical graphics.

For a brief introduction to the ideas behind the library, you can read the [introductory notes](https://seaborn.pydata.org/introduction.html#introduction). Visit the [installation page](https://seaborn.pydata.org/installing.html#installing) to see how you can download the package. You can browse the [example gallery](https://seaborn.pydata.org/examples/index.html#example-gallery) to see what you can do with seaborn, and then check out the [tutorial](https://seaborn.pydata.org/tutorial.html#tutorial) and [API reference](https://seaborn.pydata.org/api.html#api-ref) to find out how.

To see the code or report a bug, please visit the [github repository](https://github.com/mwaskom/seaborn). General support issues are most at home on [stackoverflow](https://stackoverflow.com/), where there is a seaborn tag.

Regular expression:

This module provides regular expression matching operations similar to those found in Perl.

Both patterns and strings to be searched can be Unicode strings ([str](https://docs.python.org/3/library/stdtypes.html#str)) as well as 8-bit strings ([bytes](https://docs.python.org/3/library/stdtypes.html#bytes)). However, Unicode strings and 8-bit strings cannot be mixed: that is, you cannot match a Unicode string with a byte pattern or vice-versa; similarly, when asking for a substitution, the replacement string must be of the same type as both the pattern and the search string.

Regular expression syntax:

A regular expression (or RE) specifies a set of strings that matches it; the functions in this module let you check if a particular string matches a given regular expression (or if a given regular expression matches a particular string, which comes down to the same thing).

Example:

INPUT:- **import** **re**

**>>>** m = re.search('(?<=abc)def', 'abcdef')

**>>>** m.group(0)

OUTPUT:- 'def'

# Natural Language Toolkit:

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to [over 50 corpora and lexical resources](http://nltk.org/nltk_data/) such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active [discussion forum](http://groups.google.com/group/nltk-users).

Thanks to a hands-on guide introducing programming fundamentals alongside topics in computational linguistics, plus comprehensive API documentation, NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.

CODES

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

dataset=pd.read\_csv('C:\\Users\\acer\\Desktop\\shortrev.csv')

dataset['Text'][2]

import re

import nltk

from nltk.corpus import stopwords

from nltk.stem.porter import PorterStemmer

ps=PorterStemmer()

dataset['Text'][0]

clean\_review=[]

for i in range(5000):

text=re.sub('[^a-zA-Z]', ' ',dataset['Text'][i])

text=text.lower()

text=text.split()

text=[ps.stem(word) for word in text if not word in set(stopwords.words('english'))]

text=" ".join(text)

clean\_review.append(text)

from sklearn.feature\_extraction.text import CountVectorizer

cv=CountVectorizer(max\_features=400)

X=cv.fit\_transform(clean\_review)

X=X.toarray()

dataset['label'] = np.nan

for i in range(0,5000):

if( dataset['Score'][i]>=3):

dataset['label'][i]=1

else:

dataset['label'][i]=0

y=dataset['label'].values

############## Naive Bayes Algorithm ###############

from sklearn.naive\_bayes import GaussianNB

nb=GaussianNB()

nb.fit(X,y)

nb.score(X,y)

############# Logistic Regression #################

from sklearn.linear\_model import LogisticRegression

log\_reg = LogisticRegression()

log\_reg.fit(X,y)

log\_reg.score(X,y)

######### Printing all the relevant features for classification ########

print(cv.get\_feature\_names())

OUTPUT

In[1]:import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

In[2]:dataset=pd.read\_csv('C:\\Users\\acer\\Desktop\\shortrev.csv')

dataset['Text'][2]

Out[2]: 'This is a confection that has been around a few centuries. It is a light, pillowy citrus gelatin with nuts - in this case Filberts. And it is cut into tiny squares and then liberally coated with powdered sugar. And it is a tiny mouthful of heaven. Not too chewy, and very flavorful. I highly recommend this yummy treat. If you are familiar with the story of C.S. Lewis\' "The Lion, The Witch, and The Wardrobe" - this is the treat that seduces Edmund into selling out his Brother and Sisters to the Witch.'

In[3]:import re

import nltk

from nltk.corpus import stopwords

from nltk.stem.porter import PorterStemmer

ps=PorterStemmer()

dataset['Text'][0]

clean\_review=[]

In[4]:for i in range(5000):

text=re.sub('[^a-zA-Z]', ' ',dataset['Text'][i])

text=text.lower()

text=text.split()

text=[ps.stem(word) for word in text if not word in set(stopwords.words('english'))]

text=" ".join(text)

clean\_review.append(text)

In[5]:from sklearn.feature\_extraction.text import CountVectorizer

cv=CountVectorizer(max\_features=400)

X=cv.fit\_transform(clean\_review)

X=X.toarray()

In[6]:dataset['label'] = np.nan

for i in range(0,5000):

if( dataset['Score'][i]>=3):

dataset['label'][i]=1

else:

dataset['label'][i]=0

y=dataset['label'].values

\_\_main\_\_:4: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy

\_\_main\_\_:6: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/indexing.html#indexing-view-versus-copy

In[7]:from sklearn.linear\_model import LogisticRegression

log\_reg = LogisticRegression()

log\_reg.fit(X,y)

log\_reg.score(X,y)

######### Printing all the relevant features for classification ########

print(cv.get\_feature\_names())

Out[7]:['abl', 'absolut', 'actual', 'ad', 'add', 'addit', 'aftertast', 'ago', 'almost', 'also', 'altern', 'although', 'alway', 'amaz', 'amazon', 'amount', 'anoth', 'anyon', 'anyth', 'around', 'arriv', 'artifici', 'avail', 'away', 'babi', 'back', 'bad', 'bag', 'bake', 'bar', 'bean', 'believ', 'best', 'better', 'big', 'biscuit', 'bisquick', 'bit', 'bitter', 'black', 'blend', 'bodi', 'bottl', 'bought', 'box', 'br', 'brand', 'bread', 'brew', 'brown', 'butter', 'buy', 'cake', 'call', 'calori', 'came', 'can', 'candi', 'care', 'carri', 'case', 'cat', 'chang', 'cheaper', 'chees', 'chicken', 'chip', 'chocol', 'choic', 'cocoa', 'coconut', 'coffe', 'color', 'com', 'come', 'compani', 'compar', 'contain', 'conveni', 'cook', 'cooki', 'corn', 'cost', 'could', 'coupl', 'cracker', 'cream', 'crisp', 'crunchi', 'cup', 'cut', 'dark', 'daughter', 'day', 'deal', 'decaf', 'decid', 'definit', 'delici', 'diet', 'differ', 'disappoint', 'dog', 'dri', 'drink', 'easi', 'eat', 'egg', 'either', 'els', 'end', 'energi', 'enjoy', 'enough', 'especi', 'etc', 'even', 'ever', 'everi', 'everyon', 'everyth', 'excel', 'expect', 'expens', 'extra', 'fact', 'famili', 'fan', 'far', 'fast', 'fat', 'favorit', 'feed', 'feel', 'fill', 'find', 'fine', 'first', 'flavor', 'flour', 'food', 'found', 'free', 'fresh', 'friend', 'fruit', 'full', 'gave', 'get', 'gf', 'gift', 'give', 'glad', 'gluten', 'go', 'good', 'got', 'gp', 'great', 'green', 'groceri', 'ground', 'half', 'hand', 'happi', 'hard', 'health', 'healthi', 'healthier', 'help', 'high', 'highli', 'home', 'honey', 'hope', 'hot', 'hous', 'howev', 'href', 'http', 'husband', 'ice', 'includ', 'ingredi', 'instead', 'issu', 'item', 'juic', 'keep', 'kettl', 'keurig', 'kid', 'kind', 'know', 'label', 'larg', 'last', 'least', 'leav', 'less', 'let', 'licoric', 'light', 'like', 'list', 'littl', 'live', 'local', 'long', 'look', 'lot', 'love', 'low', 'made', 'make', 'mani', 'market', 'may', 'mayb', 'meal', 'meat', 'might', 'milk', 'minut', 'miss', 'mix', 'money', 'month', 'morn', 'mouth', 'much', 'must', 'natur', 'need', 'never', 'new', 'newman', 'next', 'nice', 'normal', 'noth', 'notic', 'nut', 'oil', 'ok', 'old', 'one', 'open', 'order', 'oreo', 'organ', 'origin', 'other', 'ounc', 'oz', 'pack', 'packag', 'packet', 'pancak', 'past', 'pay', 'peanut', 'peopl', 'pepper', 'per', 'perfect', 'person', 'piec', 'pineappl', 'place', 'plain', 'plastic', 'pleas', 'plu', 'pod', 'pop', 'popcorn', 'potato', 'pound', 'powder', 'prefer', 'pretti', 'price', 'probabl', 'problem', 'product', 'protein', 'purchas', 'put', 'qualiti', 'quick', 'quit', 'rather', 'read', 'real', 'realli', 'reason', 'receiv', 'recent', 'recip', 'recommend', 'regular', 'review', 'rice', 'rich', 'right', 'roast', 'run', 'said', 'salt', 'salti', 'satisfi', 'sauc', 'save', 'say', 'season', 'second', 'see', 'seem', 'serv', 'set', 'sever', 'ship', 'side', 'sinc', 'singl', 'size', 'small', 'smell', 'smooth', 'snack', 'sodium', 'soft', 'someth', 'son', 'soup', 'spice', 'spici', 'star', 'start', 'stick', 'still', 'stop', 'store', 'strong', 'stuff', 'sugar', 'super', 'sure', 'surpris', 'sweet', 'sweeten', 'syrup', 'take', 'tast', 'tasti', 'tea', 'tell', 'textur', 'thank', 'thin', 'thing', 'think', 'though', 'thought', 'three', 'time', 'tomato', 'took', 'top', 'treat', 'tri', 'turn', 'two', 'type', 'us', 'use', 'usual', 'varieti', 'vinegar', 'waffl', 'want', 'water', 'way', 'week', 'weight', 'well', 'went', 'wheat', 'white', 'whole', 'wish', 'without', 'wonder', 'work', 'worth', 'would', 'www', 'year', 'yet']

Summary

**In this** we have used [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing) to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to [voice of the customer](https://en.wikipedia.org/wiki/Voice_of_the_customer) materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from [marketing](https://en.wikipedia.org/wiki/Marketing) to [customer service](https://en.wikipedia.org/wiki/Customer_relationship_management) to clinical medicine.

References

[1]https://www.kaggle.com/nulldata/begin-your-data-analysis-in-python-with-ipl-data

[2]https://towardsdatascience.com/analysing-ipl-data-to-begin-data-analytics-with-python-5d2f610126a

[3]https://github.com/amrrs/iq18\_workshop/blob/master/data\_analysis/Data%20Analysis%20in%20Python%20with%20IPL%20Dataset.ipynb

[4] https://www.meetup.com/BangPypers/events/qxlhwkyxfbwb/