**ABSTRACT**

Sentiment analysis is a huge volume increasing at a humongous rate everyday which has made it almost impossible to evaluate the data manually. People share their opinion as in a huge number of their prevalence. In order to make the process of analysing the text automatic there are various machine learning techniques that could be applied. The data set is for those enthusiasts who are willing to play with text data and perform sentiment analysis or text classification. The huge quantity of data in textual is generated every day has no value unless processed. The text data issue can be resolved by

adopting Data mining technique using r tool. Our experimental work intends to adopt Naïve Bayes classifier is data mining techniques for the effective prediction of text

data. This data set consists of actual reviews from real people. So this data set will give a real time experience as to how to deal with textual data**.**One of the most effective tools any restaurant has is the ability to track food and beverage sales daily. Currently, Recommender systems play an important role in both academia and industry. These are very helpful for managing information overload. In this paper, we applied machine learning techniques for user reviews and analyse valuable information in the reviews. Reviews are useful for making decisions for both customers and owners. We build a machine learning model with Natural Language Processing techniques that can capture the user's opinions from users’ reviews. For

experimentation, the python language was used.

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No** | **Topic** | | |
| 1 | Introduction | | |
|  |  | | |
|  | 1.1 |  | Company Profile |
|  | 1.2 |  | Introduction to project |
|  |  | 1.2.1 | Aim Of the Project |
|  |  | 1.2.2 | Problem Definition |
| 2 | Import Dataset | | |
|  | 2.1 |  | Importing necessary Packages |
|  | 2.2 |  | Dataset |
| 3 | Text Processing | | |
| 4 | Approach | | |
| 5 | Model Training Using MultinomialNB | | |
| 6 | Model Performance | | |
| 7 | Technical use. | | |
| 8 | Deployment Using GUI | | |
| 9 | Result | | |
| 10 | Conclusion | | |
| 11 | Future Scope | | |
| 12 | Reference | | |

* 1. **Introduction**

Restaurant customers give their ratings and write reviews based on their satisfactionlevels. These ratings and reviews help the other customers to make a decision on going tothose restaurants. These ratings are also helpful for the restaurant owners to makechanges based on their reviews for improving their business Restaurant reviews containstextual information. But most machine learning algorithms work with numericaldata only. Machine learning can be considered one of the applications of artificialintelligence (AI).ML provides a way to learn the systems without being explicitlyprogrammed and this learning can be used for solving problems. Machine learning takesdata as input and it learns some important relations from data to make decisions as peruser requirements. The learning process starts with the observations like samples, and directexperience and then finding patterns in that data to make better decisions to predict orclassify new things in the future. For text, processing machine learning provides Naturalprocessing (NLP) capabilities. We can easily analyze our textual datasetsthrough NLP methodologies. NLP provides an opportunity for data analysts to applymachine learning and deep learning algorithms to our textual datasets. We make use ofmachine learning algorithms for classifying reviews and recommending the best restaurant.

**2.1 Aim of The Project –**

Normally, a lot of businesses are remained as failures due to lack of profit, lack of proper improvement measures. Mostly, restaurant owners face a lot of difficulties to improve their productivity. This project really helps those who want to increase their productivity, which in turn increases their business profits. This is the main objective of this project.

What the project does is that the restaurant owner gets to know about drawbacks of his restaurant such as most disliked food items of his restaurant by customer’s text review which is processed with ML classification algorithm (Naive Bayes)

The purpose of this analysis is to build a prediction model to predict whether a review on the restaurant is positive or negative. To do so, we will work on Restaurant Review dataset, we will load it into predictive algorithms Multinomial Naive Bayes, Bernoulli Naive Bayes and Logistic Regression. In the end, we hope to find a "best" model for predicting the review's sentiment.

**2.3 Literature Survey**

J. P. Schimberg, O. L. Haimon, G. R. Hayes, and H. Anton-Culver [2] has proposed the “Supplementing public health inspection via social media” Mining publicly available

crowd sourced data to develop a surveillance method for tracking foodborne illness risk factors gives health inspectors an improved ability to identify restaurants with greater odd so flow health code ratings and violations outside of the normal inspection window.

A. Sadilek, S. Brennan, H. Kautz, and V. Silenzio Nemesis [3] has proposed“Which restaurants should people avoid today” Computational approaches to health monitoring and

epidemiology continue to evolve rapidly. The proposed work presented an end-to-end system, nEmesis, that automatically identifies restaurants posing public health

risks. Leveraging a language model of Twitter users’ online communication, it makes the people to nEmesis finds individuals who are likely suffering from a foodborne

illness from a colony. People’s visits to restaurants are modelled by matching GPS data embedded in the messages with restaurant addresses.

C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky [5] has proposed “Stanford corenlp natural language processing toolkit” the proposed

work describe the design and use of the Stanford Core NLP toolkit, an extensible pipeline that provides core natural language analysis. This tool kit is quite widely used, both in the research NLP community and also among commercial and government users of open source NLP technology. The method suggest that it follows from a

simple, approachable and easy design, which can be straightforward interfaces, the inclusion of robust and in the good quality analysis components, and not to requiring

use of a large amount of associated baggage. The approached method defines the design and development of stanford Core NLP, and gives the common core natural

language processing steps, from the tokenization.

K. Lee, A. Agrawal, and A. N. Choudhary [6] has proposed “Mining social media streams to improve public health allergy surveillance” Allergies are one of the most common chronic diseases worldwide. One in five Americans suffer from either allergy or asthma symptoms. With the prevalence of social media, people sharing experiences and opinions on personal health symptoms and concerns on social media are increasing. Allergy is the fifth most

common chronic diseases in the United States1. The complexity and severity of allergic diseases are increasing worldwide. One in five Americans have either allergy or

asthma symptoms. In 2012, 7.5% of adults (17.6 million adults) and 9% of children (6.6 million children) were diagnosed with hay fever. Continuous use of allergy

medication can worsen patients’ health conditions and lead to side effects and other serious medical complications.

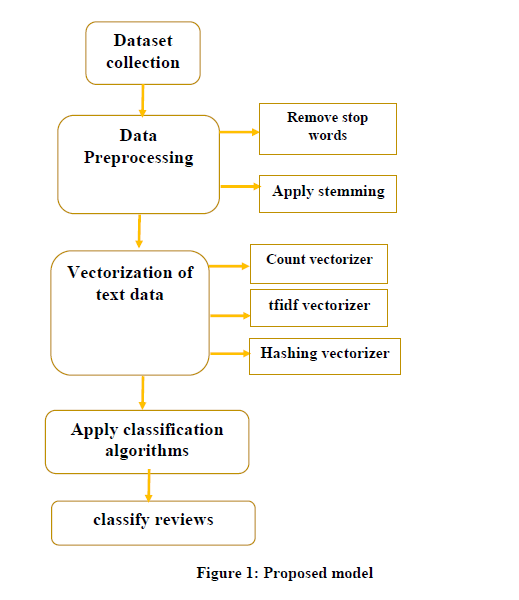
K. Lee, A. Agrawal, and A. Choudhary [7] has proposed the “Real-time disease surveillance using twitter data: demonstration on flu and cancer” Social media is producing massive amounts of data on an unprecedented scale. Here people share their experiences and opinions on various topics, including personal health issues, symptoms, treatments, side-effects, and so on. The proposed work do he publicly available social media data an invaluable

resource for mining available and wanted actionable healthcare in the media. In this paper, we describe a novel real-time flu and cancer surveillance system that uses

spatial, temporal and text mining on Twitter data.

* 1. **Research Methodology**

**Structure of Proposed model:**



Generally, we need a procedure for representing text information for the ML algorithm.Bag-of-words are useful to complete this task. This model is simple to implement. It is oneof the methods to extract features from the given text for machine learning models. The Bagof Words model is used to pre-process the input text by changing it into a bag ofwords. The bow can be represented using a table, which contains the count of wordscorresponding to the word itself.

* 1. **Approach**

**3.2.1 Tools & Technologies Used:**

NLTK

Machine Learning

Python

Tkinter

Mysql

Pandas

**Description:**

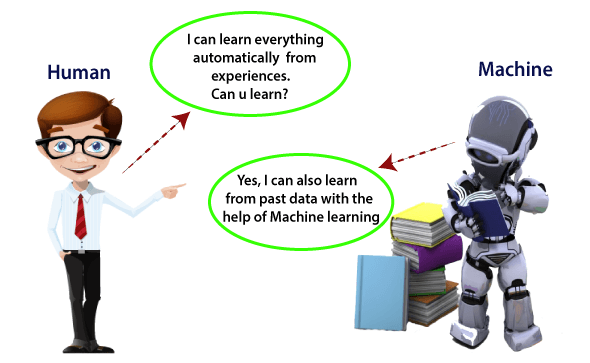
**Python:**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms and can be freely distributed.

**Machine Learning:**

Machine learning (ML) is a type of artificial intelligence ([AI](https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence)) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning [algorithms](https://www.techtarget.com/whatis/definition/algorithm) use historical data as input to predict new output values.

Machine learning is important because it gives enterprises a view of trends in customer behaviour and business operational patterns, as well as supports the development of new products. Many of today's leading companies, such as Facebook, Google and Uber, make machine learning a central part of their operations. Machine learning has become a significant competitive differentiator for many companies



## Need for Machine Learning

The need for machine learning is increasing day by day. The reason behind the need for machine learning is that it is capable of doing tasks that are too complex for a person to implement directly. As a human, we have some limitations as we cannot access the huge amount of data manually, so for this, we need some computer systems and here comes the machine learning to make things easy for us.

We can train machine learning algorithms by providing them the huge amount of data and let them explore the data, construct the models, and predict the required output automatically. The performance of the machine learning algorithm depends on the amount of data, and it can be determined by the cost function. With the help of machine learning, we can save both time and money.

The importance of machine learning can be easily understood by its uses cases, Currently, machine learning is used in **self-driving cars**, **cyber fraud detection**, **face recognition**, and **friend suggestion by Facebook**, etc. Various top companies such as Netflix and Amazon have build machine learning models that are using a vast amount of data to analyze the user interest and recommend product accordingly.

Classification of Machine Learning

At a broad level, machine learning can be classified into three types:

1. **Supervised learning**
2. **Unsupervised learning**
3. **Reinforcement learning**



**Machine Learning Life cycle:**

****

**Pandas:-**

**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](https://en.wikipedia.org/wiki/Data_analysis). In particular, it offers [data structures](https://en.wikipedia.org/wiki/Data_structure) 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 "[**pan**el **da**ta](https://en.wikipedia.org/wiki/Panel_data)", an [econometrics](https://en.wikipedia.org/wiki/Econometrics) term for [data sets](https://en.wikipedia.org/wiki/Data_set) that include observations over multiple time periods for the same individuals.[[3]](https://en.wikipedia.org/wiki/Pandas_(software)#cite_note-3) Its name is a play on the phrase "Python data analysis" itself. [Wes McKinney](https://en.wikipedia.org/wiki/Wes_McKinney) started building what would become pandas at [AOR](https://en.wikipedia.org/wiki/AQR_Capital) while he was a researcher there from 2007 to 2010.

Library features

* DataFrame [object](https://en.wikipedia.org/wiki/Object-oriented_programming) for data manipulation with integrated indexing.
* Tools for reading and writing data between in-memory [data structures](https://en.wikipedia.org/wiki/Data_structure) and different [file formats](https://en.wikipedia.org/wiki/File_format).
* Data alignment and integrated handling of missing data.
* Reshaping and pivoting of data sets.
* Label-based slicing, fancy indexing, and subsetting of large data sets.
* Data structure column insertion and deletion.
* Group by engine allowing split-apply-combine operations on data sets.
* Data set merging and joining.

**Matplotlib and Seaborn-:**

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of [matplotlib](https://www.geeksforgeeks.org/python-introduction-matplotlib/) library and also closely integrated to the data structures from [pandas](https://www.geeksforgeeks.org/introduction-to-pandas-in-python/).  
Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset-oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset.

## Different categories of plot in Seaborn

Plots are basically used for visualizing the relationship between variables. Those variables can be either be completely numerical or a category like a group, class or division. Seaborn divides plot into the below categories – 

* **Relational plots:** This plot is used to understand the relation between two variables.
* [**Categorical plots:**](https://www.geeksforgeeks.org/seaborn-categorical-plots/)This plot deals with categorical variables and how they can be visualized.
* [**Distribution plots:**](https://www.geeksforgeeks.org/seaborn-distribution-plots/)This plot is used for examining univariate and bivariate distributions
* [**Regression plots:**](https://www.geeksforgeeks.org/seaborn-regression-plots/)The regression plots in seaborn are primarily intended to add a visual guide that helps to emphasize patterns in a dataset during exploratory data analyses.
* [**Matrix plots:**](https://www.geeksforgeeks.org/ml-matrix-plots-in-seaborn/) A matrix plot is an array of scatterplots.
* **Multi-plot grids:**It is an useful approach is to draw multiple instances of the same plot on different subsets of the dataset.

**Regular Expression:**

A RegEx, or Regular Expression, is a sequence of characters that forms a search pattern.RegEx can be used to check if a string contains the specified search pattern. Python has a built-in package called re, which can be used to work with Regular Expressions. Regular expressions are a powerful language for matching text patterns. This page gives a basic introduction to regular expressions themselves sufficient for our Python exercises and shows how regular expressions work in Python. The Python "re" module provides regular expression support

**Scikit learn:**

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

## Features

Rather than focusing on loading, manipulating and summarising data, Scikit-learn library is focused on modeling the data. Some of the most popular groups of models provided by Sklearn are as follows −

**Supervised Learning algorithms** − Almost all the popular supervised learning algorithms, like Linear Regression, Support Vector Machine (SVM), Decision Tree etc., are the part of scikit-learn.

**Unsupervised Learning algorithms** − On the other hand, it also has all the popular unsupervised learning algorithms from clustering, factor analysis, PCA (Principal Component Analysis) to unsupervised neural networks.

**Clustering** − This model is used for grouping unlabeled data.

**Cross Validation** − It is used to check the accuracy of supervised models on unseen data.

**Dimensionality Reduction** − It is used for reducing the number of attributes in data which can be further used for summarisation, visualisation and feature selection.

**Ensemble methods** − As name suggest, it is used for combining the predictions of multiple supervised models.

**Feature extraction** − It is used to extract the features from data to define the attributes in image and text data.

**Feature selection** − It is used to identify useful attributes to create supervised models.

**Open Source** − It is open source library and also commercially usable under BSD licens

**3.2.2 Implementation**

**Importing necessary packages**:

import pandas as pd

import re

import nltk

import matplotlib.pyplot as plt

import seaborn as sns

import sklearn

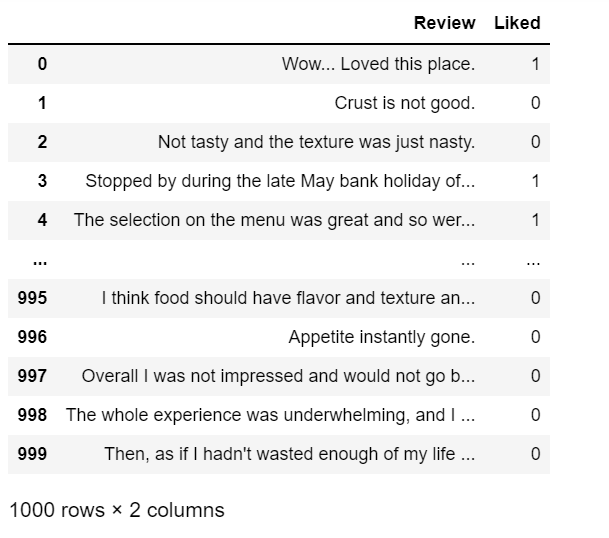
* + 1. **Dataset Description-:**

**Import Dataset**

The dataset is downloaded from superdatascience.com. The dataset contains reviews in the

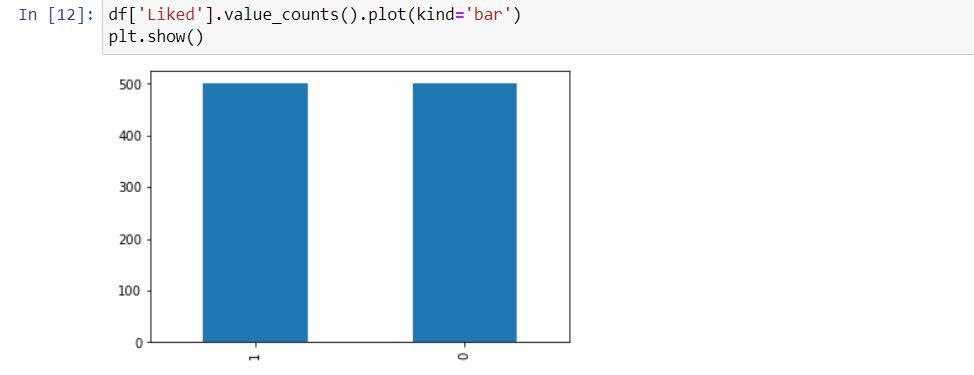
text format. It also contains a feature that indicates whether the review is positive or negative.

A positive review can de indicate by 1 and a negative review is indicated by 0. This dataset contains 1000 reviews

****

The review is positive or negative. 1 for positive review and 0 for negative review

This dataset contains 1000 reviews that is balanced with 500 for positive and 500 for negatives

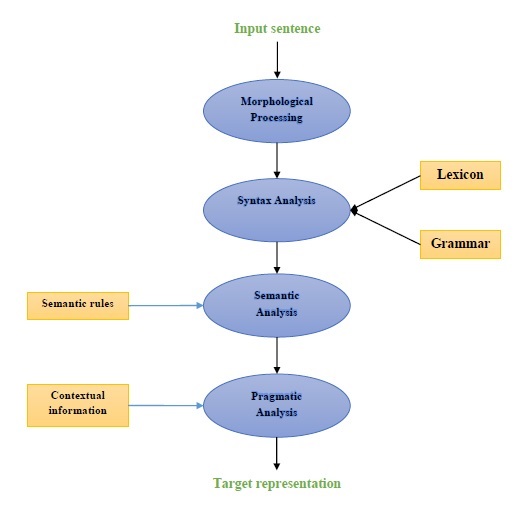
****

* + 1. **Text Processing by using NLTK:-**

Language is a method of communication with the help of which we can speak, read and write. Natural Language Processing (NLP) is the sub field of computer science especially Artificial Intelligence (AI) that is concerned about enabling computers to understand and process human language. We have various open-source NLP tools but NLTK (Natural Language Toolkit) scores very high when it comes to the ease of use and explanation of the concept. The learning curve of Python is very fast and NLTK is written in Python so NLTK is also having very good learning kit. NLTK has incorporated most of the tasks like tokenization, stemming, Lemmatization, Punctuation, Character Count, and Word count. It is very elegant and easy to work with.

NLTK consists of the most common algorithms such as tokenizing, part-of-speech tagging, stemming, sentiment analysis, topic segmentation, and named entity recognition. NLTK helps the computer to analysis, pre-processes, and understand the written text.

NLTK is a toolkit build for working with NLP in Python. It provides us various text processing libraries with a lot of test datasets. A variety of tasks can be performed using NLTK such as tokenizing, parse tree visualization, etc… In this article, we will go through how we can set up NLTK in our system and use them for performing various NLP tasks during the text processing step.



### Morphological Processing

Morphological processing is the first component of NLP. It includes breaking of chunks of language input into sets of tokens corresponding to paragraphs, sentences and words. For example, a word like **“everyday”** can be broken into two sub-word tokens as **“every-day”**.

### Syntax analysis

Syntax Analysis, the second component, is one of the most important components of NLP. The purposes of this component are as follows −

* To check that a sentence is well formed or not.
* To break it up into a structure that shows the syntactic relationships between the different words.
* E.g. The sentences like **“The school goes to the student”** would be rejected by syntax analyzer

### Semantic analysis

Semantic Analysis is the third component of NLP which is used to check the meaningfulness of the text. It includes drawing exact meaning, or we can say dictionary meaning from the text. E.g. The sentences like “It’s a hot ice-cream.” would be discarded by semantic analyzer.

### Pragmatic analysis

Pragmatic analysis is the fourth component of NLP. It includes fitting the actual objects or events that exist in each context with object references obtained by previous component i.e. semantic analysis. E.g. The sentences like **“Put the fruits in the basket on the table”** can have two semantic interpretations hence the pragmatic analyzer will choose between these two possibilities

**Remove special character and numerical values**

Using regular expression module we can remove special character and numerical values by using substitute method

## Word replacement using regular expression

First, we are going to replace words that matches the regular expression. But for this we must have a basic understanding of regular expressions as well as python re module. In the example below, we will be replacing contraction with their expanded forms (e.g. “can’t” will be replaced with “cannot”), all that by using regular expressions.

**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.

It may be defined as the process of breaking up a piece of text into smaller parts, such as sentences and words. These smaller parts are called tokens. For example, a word is a token in a sentence, and a sentence is a token in a paragraph.

As we know that NLP is used to build applications such as sentiment analysis, QA systems, language translation, smart chatbots, voice systems, etc., hence, in order to build them, it becomes vital to understand the pattern in the text. The tokens, mentioned above, are very useful in finding and understanding these patterns. We can consider tokenization as the base step for other recipes such as stemming and lemmatization

from nltk.tokenize import sent\_tokenize, word\_tokenize

text = "Natural language processing is an exciting area."

print(sent\_tokenize(text))

**Converting into lower case**

The main reason behind to convert text into lower case is, when we are going to remove stop word from the actual data, may be can the reviews have multiple uppercase and lowercase combinations to remove all stop words from the data there are only two ways, first one, is to identify each and every word that are present in the data and then remove from the data, which will be difficult and time consuming effort. And another way to just use the corpus of stop words from NLTK and convert them similar case and same can be removed from the data.

text = re.sub(r"[^a-zA-Z0-9]", " ", text.lower())

words = text.split()

print(words)

**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.

**Stemming**

In our text we may find many words like playing, played, playfully, etc… which have a root word, play all of these convey the same meaning. So we can just extract the root word and remove the rest. Here the root word formed is called ‘stem’ and it is not necessarily that stem needs to exist and have a meaning. Just by committing the suffix and prefix, we generate the stems.

NLTK provides us with PorterStemmer LancasterStemmer and SnowballStemmer packages.

Stemming is a technique used to extract the base form of the words by removing affixes from them. It is just like cutting down the branches of a tree to its stems. For example, the stem of the words ***eating, eats, eaten*** is ***eat***.

Search engines use stemming for indexing the words. That’s why rather than storing all forms of a word, a search engine can store only the stems. In this way, stemming reduces the size of the index and increases retrieval accuracy.

from nltk.stem.porter import PorterStemmer

# Reduce words to their stems

stemmed = [PorterStemmer().stem(w) for w in words]

print(stemmed)

### Lemmatization:

We want to extract the base form of the word here. The word extracted here is called Lemma and it is available in the dictionary. We have the WordNet corpus and the lemma generated will be available in this corpus. NLTK provides us with the WordNet Lemmatizer that makes use of the WordNet Database to lookup lemmas of words.

### POS Tagging:

Part of Speech tagging is used in text processing to avoid confusion between two same words that have different meanings. With respect to the definition and context, we give each word a particular tag and process them. Two Steps are used here:

* Tokenize text (word\_tokenize).
* Apply the pos\_tag from NLTK to the above step.

Tagging, a kind of classification, is the automatic assignment of the description of the tokens. We call the descriptor s ‘tag’, which represents one of the parts of speech (nouns, verb, adverbs, adjectives, pronouns, conjunction and their sub-categories), semantic information and so on.

On the other hand, if we talk about Part-of-Speech (POS) tagging, it may be defined as the process of converting a sentence in the form of a list of words, into a list of tuples. Here, the tuples are in the form of (word, tag). We can also call POS tagging a process of assigning one of the parts of speech to the given word.

POS tagging is an important part of NLP because it works as the prerequisite for further NLP analysis as follows −

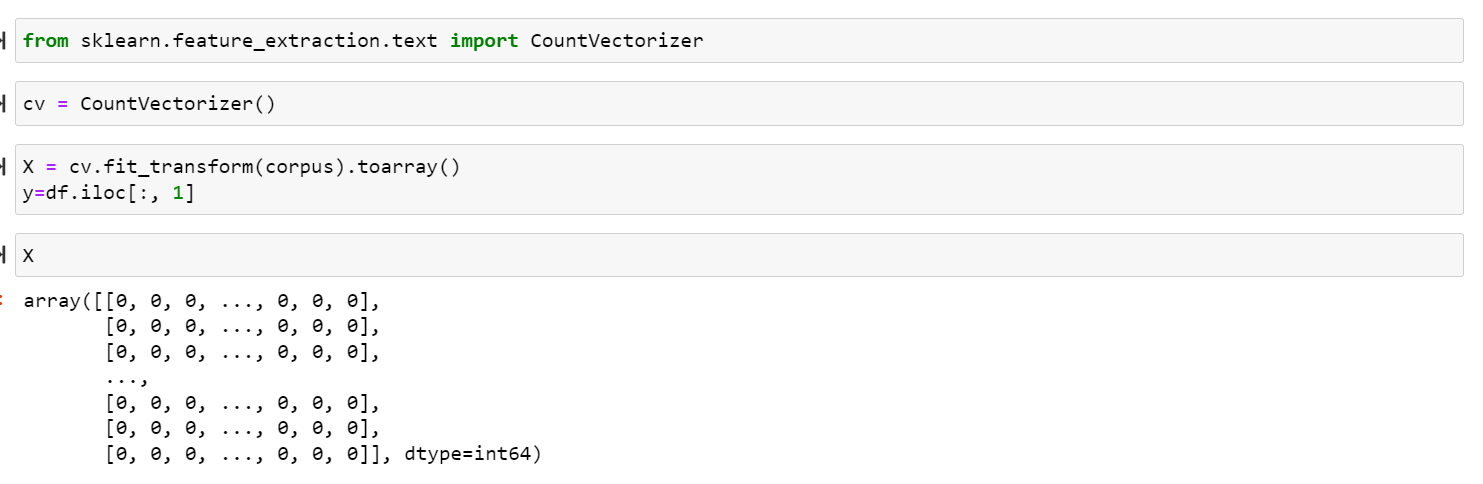
* Chunking
* Syntax Parsing
* Information extraction
* Machine Translation
* Sentiment Analysis
* Grammar analysis & word-sense disambiguation

**3.2.3 Count vectorizer**

CountVectorizer means **breaking down a sentence or any text into words by performing preprocessing tasks like converting all words to lowercase, thus removing special characters**. In NLP models can't understand textual data they only accept numbers, so this textual data needs to be vectorized.

CountVectorizer is a great tool provided by the sci-kit-learn library in Python. It is **used to transform a given text into a vector on the basis of the frequency (count) of each word that occurs in the entire text**

Machines cannot understand characters and words. So when dealing with text data we need to represent it in numbers to be understood by the machine. Countvectorizer is a method to convert text to numerical data

****

Scikit-learn’s CountVectorizer is used to convert a collection of text documents to a vector of term/token counts. It also enables the ​pre-processing of text data prior to generating the vector representation. This functionality makes it a highly flexible feature representation module for text.

**3.2.4 Train Test split**

The train-test split is a technique for evaluating the performance of a machine learning algorithm.

It can be used for classification or regression problems and can be used for any supervised learning algorithm.

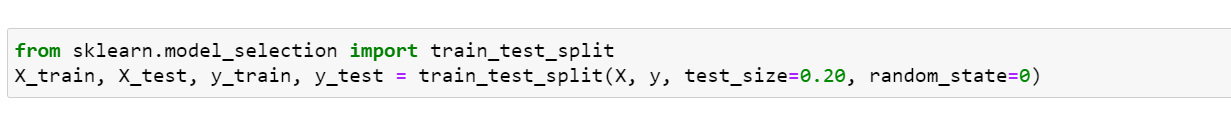
The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred to as the training dataset. The second subset is not used to train the model; instead, the input element of the dataset is provided to the model, then predictions are made and compared to the expected values. This second dataset is referred to as the test dataset.

* Train Dataset: Used to fit the machine learning model.
* Test Dataset: Used to evaluate the fit machine learning model.

The objective is to estimate the performance of the machine learning model on new data: data not used to train the model.

This is how we expect to use the model in practice. Namely, to fit it on available data with known inputs and outputs, then make predictions on new examples in the future where we do not have the expected output or target values.

The train-test procedure is appropriate when there is a sufficiently large dataset available

****

**3.2.5 Model Training**

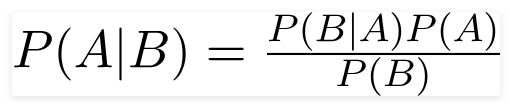
**Naive bays:**

A classifier is a machine learning model that is used to discriminate different objects based on certain features

Bayes’ theorem which was given by Thomas Bayes, a British Mathematician, in 1763 provides a means for calculating the probability of an event given some information.

# **Principle of Naive Bayes Classifier:**

A Naive Bayes classifier is a probabilistic machine learning model that’s used for classification task. The crux of the classifier is based on the Bayes theorem.

****

Naive Bayes is a supervised learning algorithm for classification so the task is to find the class of observation (data point) given the values of features.

## How Naive Bayes algorithm works?

I have a training data set of weather and corresponding target variable ‘Play’ (suggesting possibilities of playing). Now, we need to classify whether players will play or not based on weather condition

Bayes' Theorem:

* Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
* The formula for Bayes' theorem is given as:

Naïve Bayes Classifier Algorithm

**Where,**

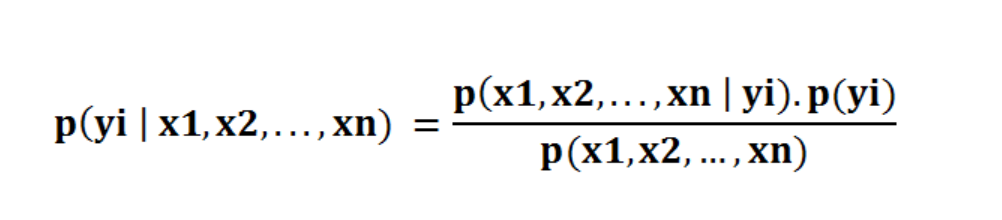
**P(A|B) is Posterior probability**: Probability of hypothesis A on the observed event B.

**P(B|A) is Likelihood probability**: Probability of the evidence given that the probability of a hypothesis is true.

**P(A) is Prior Probability**: Probability of hypothesis before observing the evidence.

**P(B) is Marginal Probability**: Probability of Evidence.

A naive Bayes classifier calculates the probability of a class given a set of feature values (i.e. p(yi | x1, x2 , … , xn)).Input this into Bayes’ theorem:



Bayes’ rule provides us with the formula for the probability of Y given some feature X. In real-world problems, we hardly find any case where there is only one feature.

When the features are independent, we can extend Bayes’ rule to what is called Naive Bayes which assumes that the features are independent that means changing the value of one feature doesn’t influence the values of other variables and this is why we call this algorithm “*NAIVE*”

# **Probability and conditional probability**

We’ve learned where the “naive” comes from. How about the “Bayes”? Bayes comes from the famous [Bayes Theorem](https://en.wikipedia.org/wiki/Bayes%27_theorem) of Thomas Bayes. To get a comprehensive understanding of Bayes’ Theorem, we should talk about probability and conditional probability first.

Bayes’ rule provides us with the formula for the probability of Y given some feature X. In real-world problems, we hardly find any case where there is only one feature.

Probability simply means the likelihood of an event to occur and always takes a value between 0 and 1 (0 and 1 inclusive). The probability of event A is denoted as **p(A)** and calculated as the number of the desired outcome divided by the number of all outcomes. For example, when you roll a die, the probability of getting a number less than three is 2 / 6. The number of desired outcomes is 2 (1 and 2); the number of total outcomes is 6.

When the features are independent, we can extend Bayes’ rule to what is called Naive Bayes which assumes that the features are independent which means changing the value of one feature doesn’t influence the values of other variables and this is why we call this algorithm “*NAIVE*”

Naive Bayes can be used for various things like face recognition, weather prediction, Medical Diagnosis, News classification, Sentiment Analysis, and a lot more.

## Assumptions of Naive Bayes

· All the variables are independent. That is if the animal is Dog that doesn’t mean that the Size will be Medium

· All the predictors have an equal effect on the outcome. That is, the animal being dog does not have more importance in deciding If we can pet him or not. All the features have equal importance.

We should try to apply the Naive Bayes formula to the above dataset however before that, we need to do some precomputations on our dataset.

**Types of Naive Bayes Classifier:**

**Multinomial Naive Bayes:**

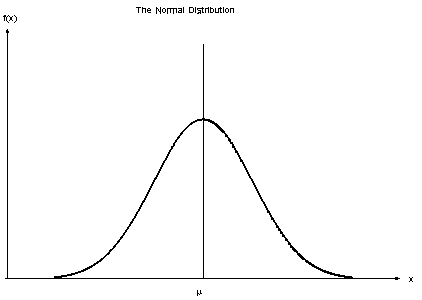
This is mostly used for document classification problem, i.e whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document.

## Bernoulli Naive Bayes:

This is similar to the multinomial naive bayes but the predictors are boolean variables. The parameters that we use to predict the class variable take up only values yes or no, for example if a word occurs in the text or not.

## Gaussian Naive Bayes:

When the predictors take up a continuous value and are not discrete, we assume that these values are sampled from a gaussian distribution.



Gaussian Distribution(Normal Distribution)

Since the way the values are present in the dataset changes, the formula for conditional probability changes to,

Naive Bayes algorithms are mostly used in sentiment analysis, spam filtering, recommendation systems etc. They are fast and easy to implement but their biggest disadvantage is that the requirement of predictors to be independent. In most of the real life cases, the predictors are dependent, this hinders the performance of the classifier

***Pros:***

* It is easy and fast to predict class of test data set. It also perform well in multi class prediction
* When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
* It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

***Cons:***

* If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
* On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict\_proba are not to be taken too seriously.
* Another limitation of [Naive Bayes](https://courses.analyticsvidhya.com/courses/naive-bayes?utm_source=blog&utm_medium=naive-bayes-explained) is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

## 4 Applications of Naive Bayes Algorithms

* **Real time Prediction:**Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.
* **Multi class Prediction:**This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.
* **Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)
* **Recommendation System:**Naive Bayes Classifier and [Collaborative Filtering](https://en.wikipedia.org/wiki/Collaborative_filtering) together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

**3.2.5.1 Logistic Regression Algorithm:**

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

**Sigmoid function/logit function:**



## Types of Logistic Regression

Generally, logistic regression means binary logistic regression having binary target variables, but there can be two more categories of target variables that can be predicted by it. Based on those number of categories, Logistic regression can be divided into following types −

### Binary or Binomial

In such a kind of classification, a dependent variable will have only two possible types either 1 and 0. For example, these variables may represent success or failure, yes or no, win or loss etc.

### Multinomial

In such a kind of classification, dependent variable can have 3 or more possible ***unordered*** types or the types having no quantitative significance. For example, these variables may represent “Type A” or “Type B” or “Type C”.

### Ordinal

In such a kind of classification, dependent variable can have 3 or more possible ***ordered*** types or the types having a quantitative significance. For example, these variables may represent “poor” or “good”, “very good”, “Excellent” and each category can have the scores like 0,1,2,3.

## Logistic Regression Assumptions

Before diving into the implementation of logistic regression, we must be aware of the following assumptions about the same −

* In case of binary logistic regression, the target variables must be binary always and the desired outcome is represented by the factor level 1.
* There should not be any multi-collinearity in the model, which means the independent variables must be independent of each other.
* We must include meaningful variables in our model.
* We should choose a large sample size for logistic regression.





## Advantages

* Logistic Regression is **one of the simplest machine learning algorithms** and is easy to implement yet provides great training efficiency in some cases. Also due to these reasons, training a model with this algorithm doesn't require high computation power.
* The predicted parameters (trained weights) give **inference about the importance of each feature**. The direction of association i.e. positive or negative is also given. So we can use logistic regression to find out the relationship between the features.
* This algorithm allows models to be **updated easily to reflect new data**, unlike decision trees or support vector machines. The update can be done using stochastic gradient descent.
* Logistic Regression **outputs well-calibrated probabilities** along with classification results. This is an advantage over models that only give the final classification as results. If a training example has a 95% probability for a class, and another has a 55% probability for the same class, we get an inference about which training examples are more accurate for the formulated problem.
* In a **low dimensional dataset** having a sufficient number of training examples, logistic regression is **less prone to over-fitting**.

## Disadvantages

* Logistic Regression is a statistical analysis model that attempts to predict precise probabilistic outcomes based on independent features. On **high dimensional datasets**, this may lead to the model being **over-fit on the training set**, which means overstating the accuracy of predictions on the training set and thus the model **may not be able to predict accurate results on the test set**. This usually happens in the case when the model is trained on little training data with lots of features. So on high dimensional datasets, Regularization techniques should be considered to avoid over-fitting (but this makes the model complex). Very high regularization factors may even lead to the model being under-fit on the training data.
* **Non linear problems can't be solved** with logistic regression **since it has a linear decision surface**. Linearly separable data is rarely found in real world scenarios. So the transformation of non linear features is required which can be done by increasing the number of features such that the data becomes linearly separable in higher dimensions.

**3.5.6 Model Performance:**

* **Accuracy score:-**

Accuracy is pretty high but this is a simple task used to show the concept and go through the steps of implementation. One important thing to remember is that the accuracy on the training set should not be much higher than the accuracy on the test set which indicates our model is too specific and not generalized well. This results in overfitting which is a serious problem for any machine learning algorithm.

* **Classification report and Confusion matrix:-**

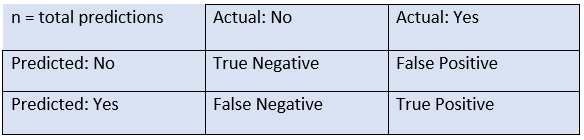
A Classification report is used to measure the quality of predictions from a classification algorithm. How many predictions are True and how many are False. More specifically, True Positives, False Positives, True negatives, and False Negatives are used to predict the metrics of a classification report as shown below. The report is copied from related to K-Means on Iris Dataset.

The report shows the main classification metrics precision, recall, and f1-score on a per-class basis. The metrics are calculated by using true and false positives, and true and false negatives. Positive and negative in this case are generic names for the predicted classes. There are four ways to check if the predictions are right or wrong:

1. **TN / True Negative:**when a case was negative and predicted negative
2. **TP / True Positive:**when a case was positive and predicted positive
3. **FN / False Negative:**when a case was positive but predicted negative
4. **FP / False Positive:**when a case was negative but predicted positive

. The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known. The matrix itself can be easily understood, but the related terminologies may be confusing. Since it shows the errors in the model performance in the form of a matrix, hence also known as an **error matrix**. Some features of Confusion matrix are given below:

* For the 2 prediction classes of classifiers, the matrix is of 2\*2 table, for 3 classes, it is 3\*3 table, and so on.
* The matrix is divided into two dimensions, that are **predicted values** and **actual values** along with the total number of predictions.
* Predicted values are those values, which are predicted by the model, and actual values are the true values for the given observations.
* It looks like the below table:



The above table has the following cases:

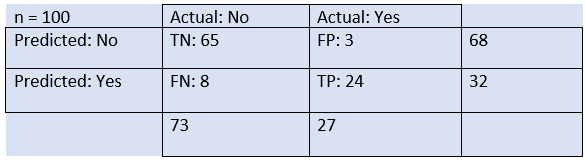
* **True Negative:** Model has given prediction No, and the real or actual value was also No.
* **True Positive:** The model has predicted yes, and the actual value was also true.
* **False Negative:** The model has predicted no, but the actual value was Yes, it is also called as **Type-II error**.
* **False Positive:** The model has predicted Yes, but the actual value was No. It is also called a **Type-I error.**

Need for Confusion Matrix in Machine learning

* It evaluates the performance of the classification models, when they make predictions on test data, and tells how good our classification model is.
* It not only tells the error made by the classifiers but also the type of errors such as it is either type-I or type-II error.
* With the help of the confusion matrix, we can calculate the different parameters for the model, such as accuracy, precision, etc.

**Example**: We can understand the confusion matrix using an example.

Suppose we are trying to create a model that can predict the result for the disease that is either a person has that disease or not. So, the confusion matrix for this is given as



From the above example, we can conclude that:

* The table is given for the two-class classifier, which has two predictions "Yes" and "NO." Here, Yes defines that patient has the disease, and No defines that patient does not has that disease.
* The classifier has made a total of **100 predictions**. Out of 100 predictions, **89 are true predictions**, and **11 are incorrect predictions**.
* The model has given prediction "yes" for 32 times, and "No" for 68 times. Whereas the actual "Yes" was 27, and actual "No" was 73 times.

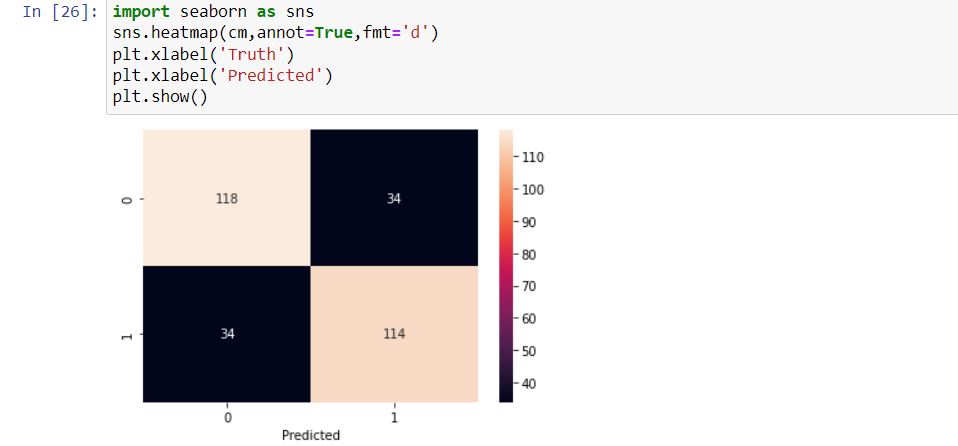
Calculations using Confusion Matrix:

We can perform various calculations for the model, such as the model's accuracy, using this matrix. These calculations are given below:

* **Classification Accuracy:** It is one of the important parameters to determine the accuracy of the classification problems. It defines how often the model predicts the correct output. It can be calculated as the ratio of the number of correct predictions made by the classifier to all number of predictions made by the classifiers. The formula
* Confusion Matrix in Machine Learning
* **Misclassification rate:** It is also termed as Error rate, and it defines how often the model gives the wrong predictions. The value of error rate can be calculated as the number of incorrect predictions to all number of the predictions made by the classifier.
* Confusion Matrix in Machine Learning
* **Precision:** It can be defined as the number of correct outputs provided by the model or out of all positive classes that have predicted correctly by the model, how many of them were actually true. It can be calculated using the below formula:  
  Confusion Matrix in Machine Learning
* **Recall:** It is defined as the out of total positive classes, how our model predicted correctly. The recall must be as high as possible.  
  Confusion Matrix in Machine Learning
* **F-measure:** If two models have low precision and high recall or vice versa, it is difficult to compare these models. So, for this purpose, we can use F-score. This score helps us to evaluate the recall and precision at the same time. The F-score is maximum if the recall is equal to the precision. It can be calculated using the below formula:
* Confusion Matrix in Machine Learning

Other important terms used in Confusion Matrix:

* **Null Error rate:** It defines how often our model would be incorrect if it always predicted the majority class. As per the accuracy paradox, it is said that "*the best classifier has a higher error rate than the null error rate.*"
* **ROC Curve:** The ROC is a graph displaying a classifier's performance for all possible thresholds. The graph is plotted between the true positive rate (on the Y-axis) and the false Positive rate (on the x-axis)

****

.

**3.5.7. Deployment using GUI and Mysql :**

Tkinter:-

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

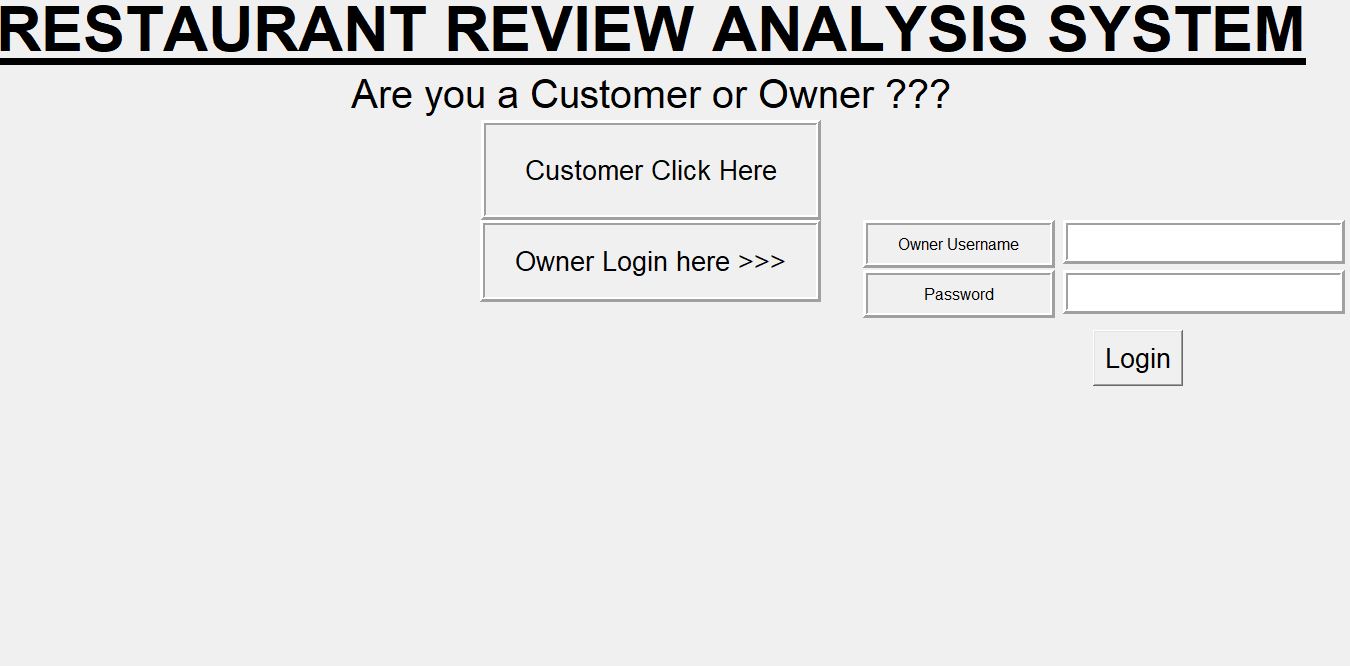
Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, Tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with Tkinter is the fastest and easiest way to create GUI applications. Creating a GUI using Tkinter is an easy task.

Database connection:-

Using mysql.connector we can connect python with MYSQL server

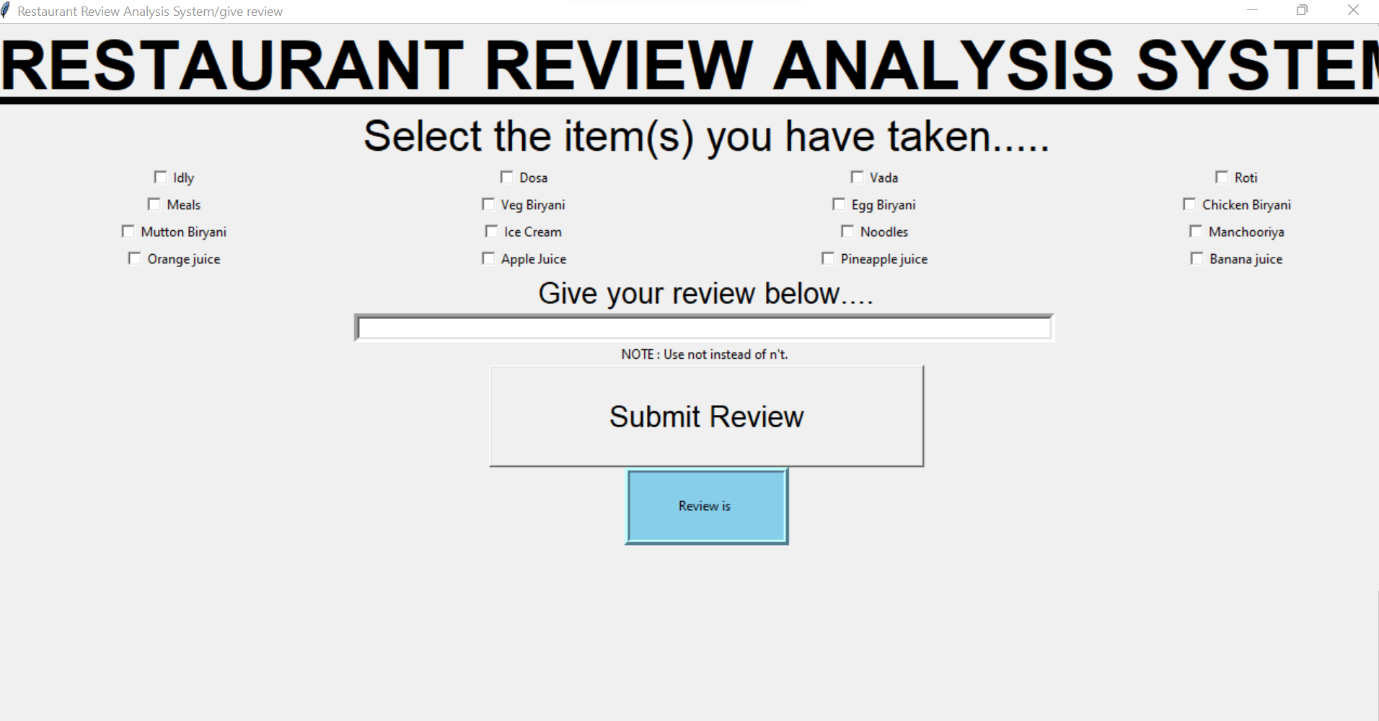
MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

****

**Customer Section:**

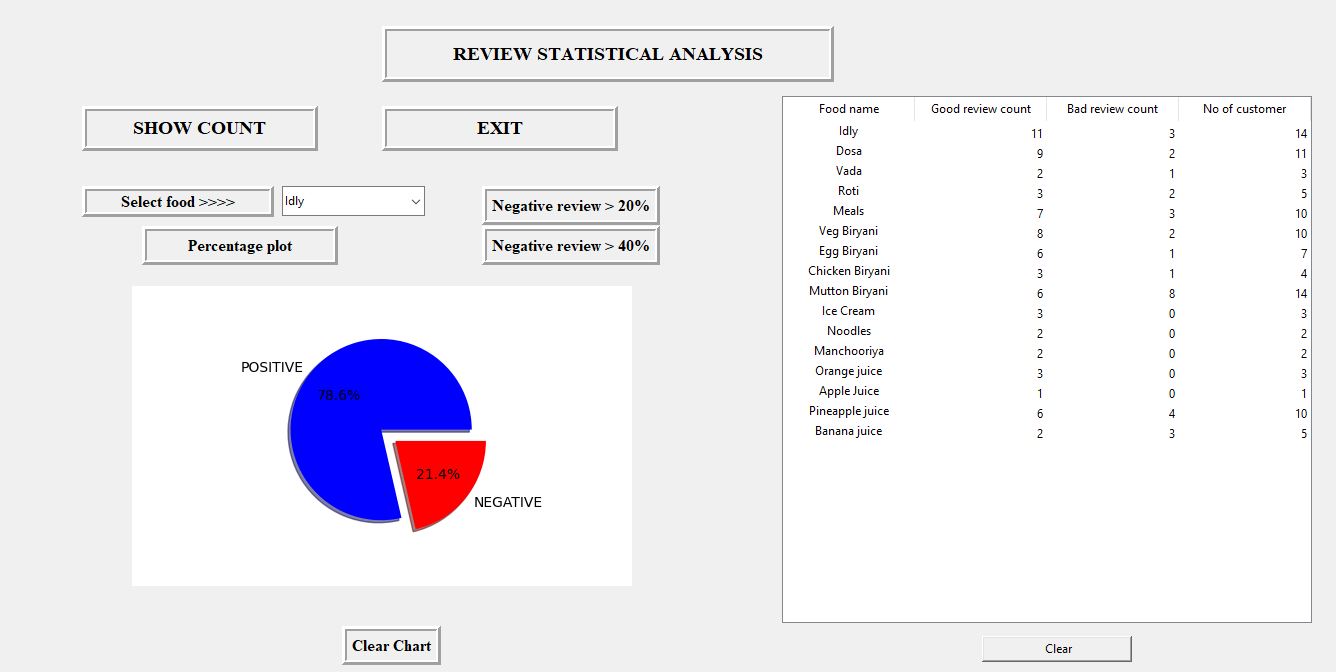
Where customer will give text review by selecting food and prediction for review whether it is good or bad is done by using Machine learning and stored the data in Mysql with food items

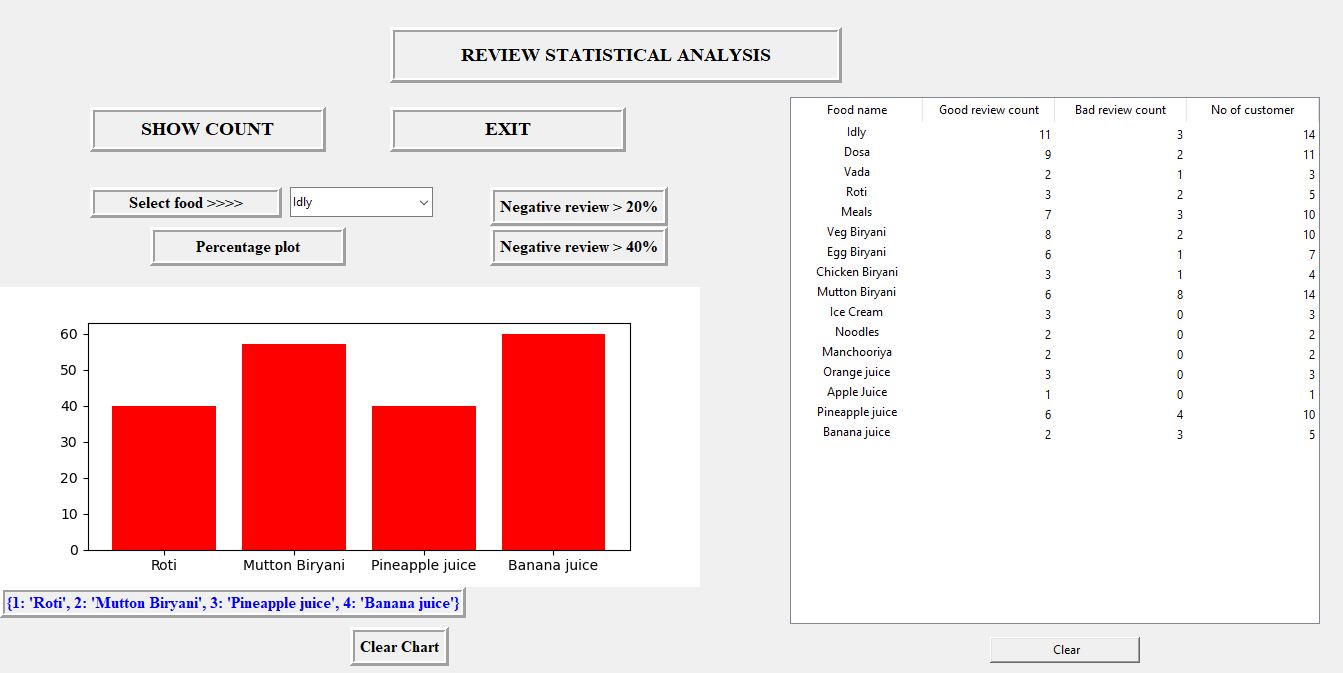
****

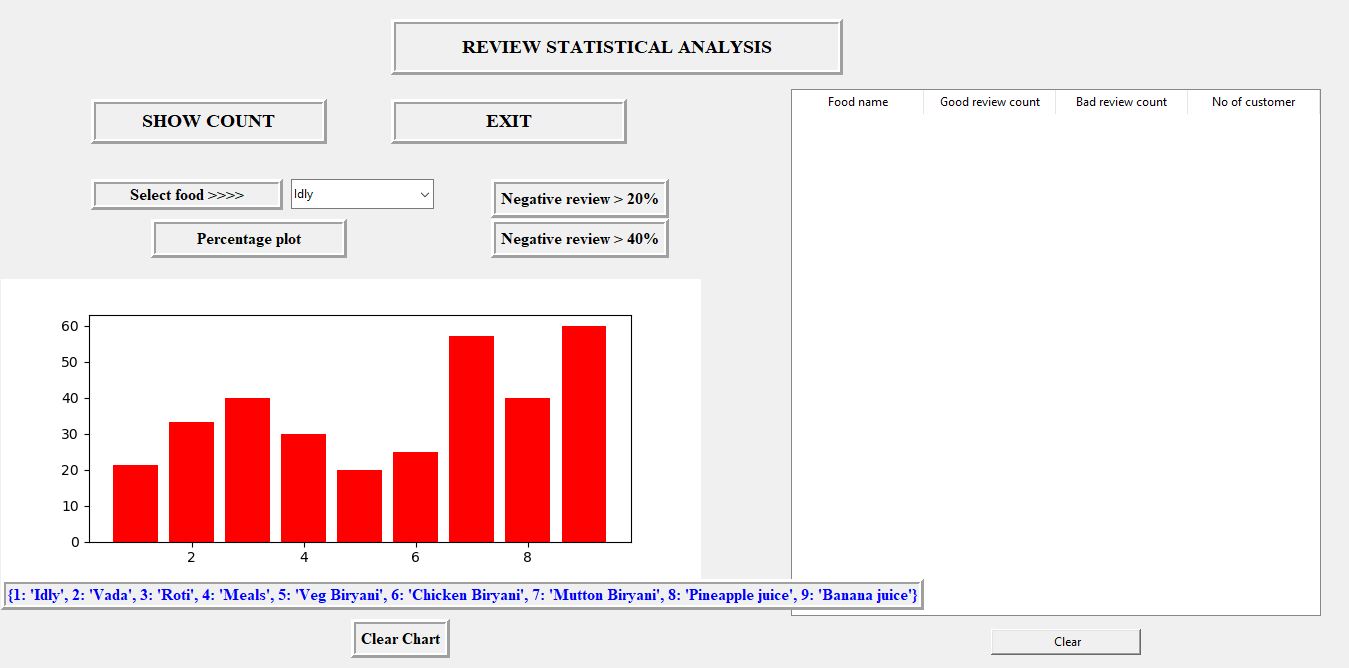
**Owner Login:**

Owner can see Stastical analysis of reviews that are stored in database with food and number of customers

We will get Count and percentage plot also the food that has greater than 20 percent negative review and also greater than 40 percent negative

****

****

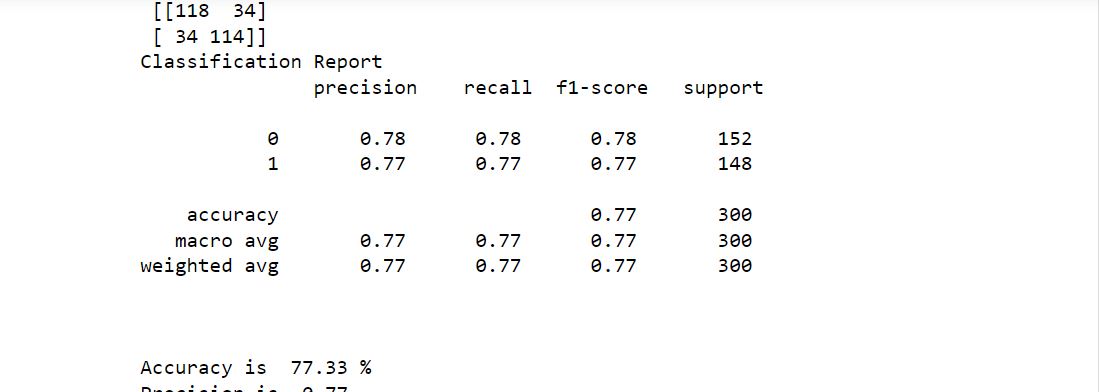
****

**Result and Analysis**

In this study, an attempt has been made to classify sentiment analysis for restaurant reviews using machine learning techniques. Two algorithms namely Multinomial Naive Bayes and Bernoulli Naive Bayes are implemented.

Evaluation metrics used here are accuracy, precision and recall.

Multinomial Naive Bayes is slightly better method compared to Bernoulli Naive Bayes and Logistic Regression, with 77.33% accuracy which means the model built for the prediction of sentiment of the restaurant review gives 77.33% right prediction.

****

**Conclusion:-**

How we landed to this work, well we were searching for a problem where we can create a model that will be useful for general public, which will help to take decisions for purchasing news product or services and it will give them a brief explanation of the product/service so that they need not to waste much time to take decision. Finally, we reached to this SemEval2014 task. In this work we have to do aspect-based sentiment analysis, for the general public and to the owner of the restaurant as well. This will help both parties. For the owner of the restaurant, it will help to find their negative and positive aspects. And they can improve their services/product quality by knowing the status of the respective aspect. Here we have four major aspects [service, food, ambiance, price] and one default category [Anecdotes/Miscellaneous] if a review does not have any aspect then it will go to default category.

**Future Work**

While calculating the sentiment, the sentiment is taken from sentiwordnet3.0, it is for the English language only, we will use different techniques and different libraries like TextBlob to get the sentiment.

This work is done only in the English language, the work can be extended to Indian languages, we have to study more research paper on sentiment analysis on Indian language, and aspect category detection for Indian language, as far as concerned about my knowledge, there are not any sentiwordnet3.0 and Wordnet for Indian language, so we need to implement the work from the scratch. It would be a great

**Reference**

* ‘**Sentiment Analysis of Restaurant Customer Reviews on TripAdvisor using Naive Bayes’, 12th International Conference on Information & Communication Technology and System (lCTS) 2019 by Kelly Rossa Sungkono,Riyanarto Sarno**
* **‘Sentiment Analysis of Restaurant Reviews’, International Research Journal of Engineering and Technology (IRJET) Pooja Tyagi Student, Dept. of Computer Science Engineering, Galgotias University, UP, India**
* **Ariyasriwatana, W., Buente, W., Oshiro, M., & Streveler, D. (2014). Categorizing health-related cues to action: using Yelp reviews of restaurants in Hawaii. New Review of Hypermedia and Multimedia, 20(4), 317-340.**
* **Identifying Restaurant Features via Sentiment Analysis on Yelp Reviews Boya Yu, Jiaxu Zhou, Yi Zhang, Yunong Cao Center for Urban Science & Progress New York University New York, NY, the United States**
* **https://github.com/abhi7585/Restaurant-Review-Sentiment-Analysis**
* **https://www.kaggle.com/code/apekshakom/sentiment-analysis-of-restaurant-reviews**
* **https://www.kaggle.com/code/apekshakom/sentiment-analysis-of-restaurant-reviews**