Sentiment Analysis

About the project

Sentiment analysis refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information.

We performed analysis on 25000 movie reviews and each review was marked with a 0 or 1 response/(sentiment) which means negative or positive response respectively. We trained our model to be able to predict on new test data. The data format of dataset followed was index in first column, attributes in second column, sentiment in third column and the last column was for the text or the review. For each model we calculated accuracy of the model and the following describes the way by which we performed the necessary tasks to get the accuracy at its best.

Link: https://colab.research.google.com/drive/1_CvvZxhHrckphQHzAlw_dXiloHVEyIpi?usp=sharing

We implemented it using:

1) **Logistic Regression CV**

2) **Multinomial Naive Bayes**

Dataset

Link: https://drive.google.com/file/d/1i4TJrwOaRR12fb1HIdNZqfTGVG-n6qUC/view?usp=sharing

1) Preprocessing

Preprocessing the given data to remove unnecessary letters and convert all upper-case letters to lower-case letters.

2) The Porter stemming algorithm (or 'Porter stemmer')

It is a process for removing the commoner morphological and in flexional endings from words in English. Its main use is as part of a term normalization process that is usually done when setting up Information Retrieval systems.

3) The Bag of Words representation

Text Analysis is a major application field for machine learning algorithms. However, the raw data, a sequence of symbols cannot be fed directly to the algorithms themselves as most of them expect numerical feature vectors with a fixed size rather than the raw text documents with variable length.

In order to address this, scikit-learn provides utilities for the most common ways to extract numerical features from text content, namely:

- **tokenizing** strings and giving an integer id for each possible token, for instance by using white-spaces and punctuation as token separators.
- **counting** the occurrences of tokens in each document.
- **normalizing** and weighting with diminishing importance tokens that occur in the majority of samples / documents.

In this scheme, features and samples are defined as follows:

- each **individual token occurrence frequency** (normalized or not) is treated as a **feature**.
- the vector of all the token frequencies for a given **document** is considered a multivariate **sample**.

A corpus of documents can thus be represented by a matrix with one row per document and one column per token (e.g. word) occurring in the corpus.

We call **vectorization** the general process of turning a collection of text documents into numerical feature vectors. This specific strategy (tokenization, counting and normalization) is called the **Bag of Words** or "Bag of n-grams" representation. Documents are described by word occurrences while completely ignoring the relative position information of the words in the document.

4) Training both the models

Logistic Regression CV (aka logit, MaxEnt) classifier - Logistic regression, despite its name, is a linear model for classification rather than regression. Logistic regression is also known in the literature as logit regression, maximum-entropy classification (MaxEnt) or the log-linear classifier. In this model, the probabilities describing the possible outcomes of a single trial are modeled using a logistic function. This implementation can fit binary, One Vs-Rest, or multinomial logistic regression with optional l1, l2 or Elastic-Net regularization.

Multinomial Naive Bayes - MultinomialNB implements the naive Bayes algorithm for multinomially distributed data, and is one of the two classic naive Bayes variants used in text classification (where the data are typically represented as word vector counts, although tf-idf vectors are also known to work well in practice). The distribution is parametrized by vectors $\theta y=(\theta y1,...,\theta yn)$ for each class y, where n is the number of features (in text classification, the size of the vocabulary) and θyi is the probability $P(xi|y)$ of feature i appearing in a sample belonging to class y.

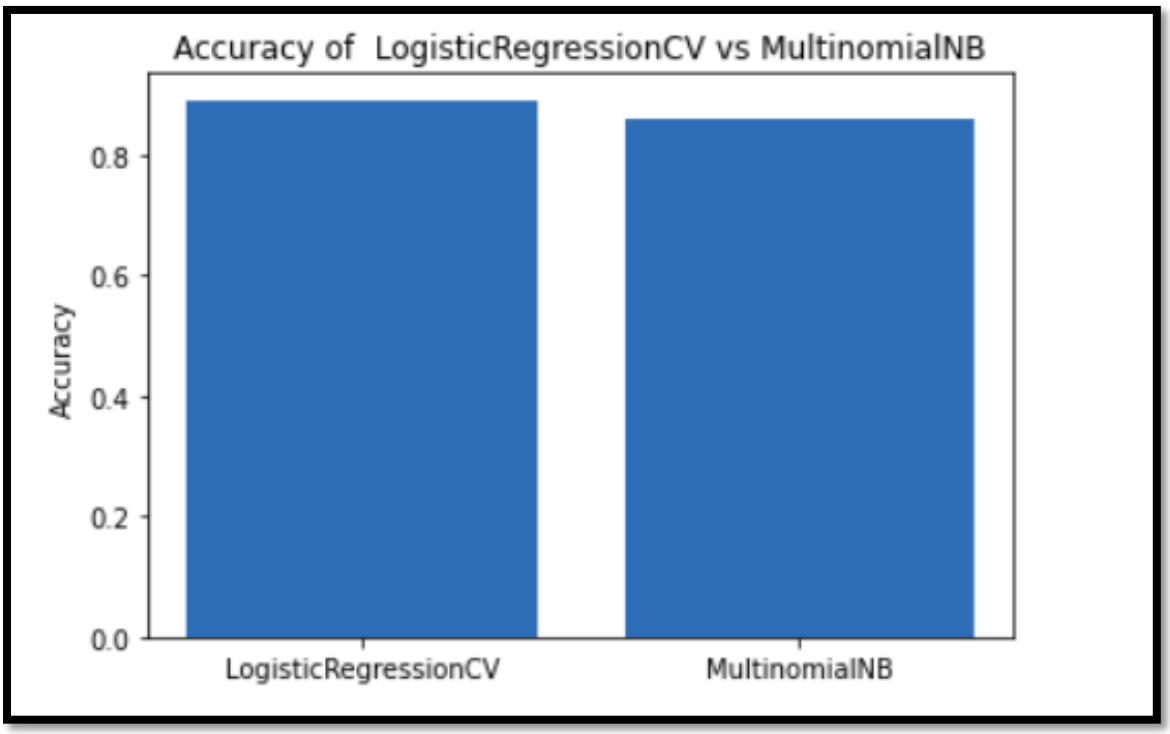
The parameters θy is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting:

$$\theta^yi=Nyi+\alpha Ny+\alpha n$$

where $Nyi=\sum x\in Txi$ is the number of times feature i appears in a sample of class y in the training set T, and $Ny=\sum i=1nNyi$ is the total count of all features for class y.

The smoothing priors $\alpha\geq 0$ accounts for features not present in the learning samples and prevents zero probabilities in further computations. Setting $\alpha=1$ is called Laplace smoothing, while $\alpha<1$ is called Lidstone smoothing.

5) Results



Accuracy of Logistic Regression CV: 89.17%

Accuracy of MultinomialNB: 86.13%

Contributors

- NAME: SAYANTAN PAL

  EMAIL: sayantan.world98@gmail.com

  PHONE: 6291110267

  ADDRESS: B – 4/7, DIAMOND PARK, JOKA, KOLKATA – 700104, WEST BENGAL

- NAME: GAURANGI SINHA

  EMAIL: gaurangi.sinha02@gmail.com

  PHONE: 8388969911

  ADDRESS: B - 261, IIT KHARAGPUR, NEAR DAV SCHOOL, KHARAGPUR,

  WEST MEDNIPUR - 721302, WEST BENGAL

- NAME: JYOTI PRAKASH DAS KARMAKAR

  EMAIL: daskarmakarj@gmail.com

  PHONE: 9800042266

  ADDRESS: 144/127/1, A.C. ROAD, INDRAPRASTHA, BERHAMPORE,

  MURSHIDAABAD, PIN – 742103, WEST BENGAL