## REVIEW CLASSIFICATION USING NLTK AND

**SPACY**

***A Mini Project report submitted to Jawaharlal Nehru Technological University, Kakinada,in the partial fulfillment for the award of the Degree in***

### MASTER OF COMPUTER APPLICATIONS

***Submitted by***

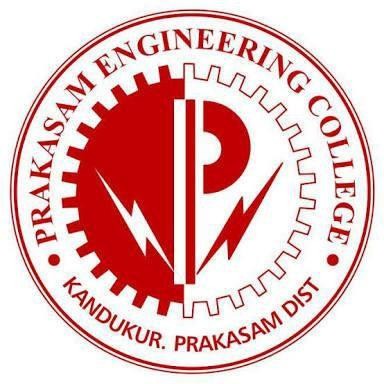
##### SIRIGIRI SAIMANIKANTA

##### (22 F91F0059)

***Under the noble guidance of***

**Mr.M M RAYUDU , M.Tech(Ph.D)**

**AssociateProfessor**

****

## PRAKASAM ENGINEERING COLLEGE

***(An ISO 9001-2008 & NAAC Accredited Institution)* (Affiliated to Jawaharlal Nehru Technological University, Kakinada)O.V.ROAD, KANDUKUR-523105, A.P.**

**2023-2024**

# PRAKASAM ENGINEERING COLLEGE

***(An ISO 9001-2008 & NAAC Accredited Institution)* (Affiliated to Jawaharlal Nehru Technological University, Kakinada)O.V.ROAD, KANDUKUR-523105, A.P.**



### DEPARTMENT OF

**MASTER OF COMPUTER APPLICATIONS BONAFIDE CERTIFICATE**

*This is to certify that the mini project entitled* **“REVIEW CLASSIFICATION USING NLTK AND SPACY”** *is a bonafide work of* **SIRIGIRI SAIMANIKANTA (22F91F0059)** *in the partial fulfillment of the requirement for the award of the degree in* **MASTER OF COMPUTER APPLICATIONS** *for the academic year 2023-2024. This work is done under my supervision and guidance.*

### Signature of the GUIDE Signature of the HOD

##### Mr. M. M. RAYUDU Mr. M. M. RAYUDU

**M.Tech(Ph.D) M.Tech, (Ph.D)**

**Signature of the External Examiner**

# DECLARATION

I do here by declare that the project work entitled **“REVIEW CLASSIFICATION USING NLTK AND SPACY”** is a genuine work carried out by me under the guidance of **Mr.M.M.RAYUDU M.Tech (Ph.D)** in partial fulfillment for the award of the degree of **“Master of Computer Applications”** of **Jawaharlal Nehru Technological University, Kakinada.**

##### SIRIGIRI SAIMANIKANTA

**(22F91F0059)**

# ACKNOWLEDGEMENT

I feel to render my thankful acknowledgement to the following distinguished personalities, who stretched their helping hand to me, in completing my mini project work.

I am very grateful and my sincere thanks to our secretary & correspondent **Dr.K.RAMAIAH** of **PRAKASAM ENGINEERING COLLEGE** for giving this opportunity.

I hereby, express my regards and extend my gratitude to our PRINCIPAL, **Dr.K.RAVI KUMAR** , for giving this opportunity to do the thesis as a part of our course.

I express my deep sense of gratitude to **Mr.M.M.RAYUDU, M.Tech(Ph.d),** Head of the Department, **Department of MCA** for having shown keen interest at every stage of development of our thesis and guiding us in every aspect.

And I am thankful to my Internal Guide **Mr.M.M.RAYUDU, M.Tech(Ph.d)** who has channeled my thoughts and timely suggestions.

I would also like to thank all my Faculties in Prakasam Engineering College for their constant encouragement and for being a great group of knowledgeable and cooperative people to work with.

##### SIRIGIRI SAIMANIKANTA

**(22F91F0059)**

**TABLE OF CONTENTS**

|  |  |
| --- | --- |
| **CHAPTER**  **ABSTRACT** | **PAGE NO.** |
| 1. **INTRODUCTION**    1. Introduction | 1 |
| 1.2 Existing System | 3 |
| 1.2.1 Drawbacks | 3 |
| 1.3 Proposed System | 3 |
| 1.3.1 Features or Objectives of Proposed | 4 |
| 1.3.2Advantages of Proposed | 4 |

#### LITERATURE SURVEY

* 1. Reference paper1 5
  2. Reference paper2 5
  3. Reference paper3 6
  4. Reference paper4 6
  5. Reference paper5 7

1. **SYSYEM ANALYSIS**

|  |  |  |
| --- | --- | --- |
| 3.1 Introduction |  | 8 |
| 3.1.1Methodology |  | 8 |
| 3.1.2.CRISP-DM | 8 |  |
| 3.1.3.Data Collection |  | 9 |
| 3.1.4.Data Preprocessing |  | 9 |
| 3.1.5.Exploratory Data Analysis |  | 10 |
| 3.2 Feasibility Study |  | 11 |
| 3.2.1Operational Feasibility |  | 12 |
| 3.2.2 Economic Feasibility |  | 13 |
| 3.3 RequirementSpecifications |  | 14 |
| 3.3.1Hardware requirements |  | 14 |
| 3.3.2 Software requirements |  | 15 |
|  |  |  |
|  |  |  |

1. **SYSTEM DESIGN**
   1. Introduction 16
      1. [Module Description 16](#_TOC_250007)
      2. [Classification by Decision Tree Induction 16](#_TOC_250006)
      3. [.Classification by Random Forest 17](#_TOC_250005)

[4.1.4.Classification by Logistic Regression 18](#_TOC_250004)

* 1. [Data Design 19](#_TOC_250003)
  2. [UML DIAGRAMS 20](#_TOC_250002)
     1. [Class Diagram 21](#_TOC_250001)
     2. UsecaseDiagram 22
     3. Sequence Diagram 23
     4. [Deployment Diagram 24](#_TOC_250000)

CONCLUSIONSANDFUTUREENHANCEMENT 25

REFFERENCES 26

**ABSTRACT**

The rise of the internet has made online reviews an increasingly valuable and significant source of information for people. Consequently, there has been a surge of interest in automatic review mining and summarizing as a prominent research area. "Text Classification" stands out as a crucial task in Natural Language Processing (NLP). It involves categorizing text strings or documents into different groups based on their content. Some instances of text classification include determining audience sentiment from social media, identifying spam and non-spam emails, tagging customer queries automatically, and classifying blog posts into various categories. This particular project concentrates on a specific domain, which is Movie Reviews. It employs a multi-knowledge-based approach that combines statistical analysis with movie knowledge. The dataset utilized in this project is the Movie Review Dataset, sourced from the IMDB movie reviews dataset. This dataset comprises thousands of positive and negative movie reviews. To ensure consistency, the dataset undergoes various cleaning and preprocessing techniques, such as converting all reviews to English language. The experimental findingsdemonstrate the effectiveness of the multi-knowledge-based approach in mining and summarizing movie reviews, thereby providing the audience with a concise and comprehensive final assessment of the movie.

I

**CHAPTER1**

### INTRODUCTION

**1.1.INTRODUCTION**

Review Classification, also known as Sentiment Analysis, involves categorizing the viewpoint or review expressed in a text using techniques from information retrieval and computational linguistics. Rather than focusing on the topic itself, the emphasis is placed on the significance of the review. Sentiment analysis utilizes methods such as natural language processing and text analytics to extract subjective information from source materials like reviews. Reviews play a crucial role in our decision-making process by providing valuable insights. Online review sites and personal blogs leverage information technologies to gather sentiments about products or objects. The primary objective of Review Classification is to determine the polarity of comments (positive, negative, or neutral) by identifying the features and components of the object that have been commented on in each document.

In the realm of Review Classification studies, the focus is on the economic impact resulting from reviews, as well as concerns regarding privacy breaches. These studies examine the nature of reviews, which can be categorized as either direct reviews or comparative reviews. Direct reviews express sentiments towards specific targets such as products, events, topics, or individuals. For example, a direct review might state, "The climax of the movie is gripping." On the other hand, comparative reviews highlight the similarities or differences

between multiple objects and often establish an order of preference. For instance, a comparative review might state, "Movie X is better than movie Y." Comparative reviews can further be classified into different types, including Non-equal Gradable (comparisons using "less than"), Equative (comparisons using "same"), and Superlative (comparisons using "longest").As of now, there are eight teams that compete with one another in a double round- robin fashion during the league stage. After the league stage, the top four teams in the league points table qualify to the playoffs.

Review Classification can be conducted either at the document level or at the sentence level. At the sentence level, there are two tasks involved: subjective classification and objective classification.

Objective classification is applied to sentences that provide factual information without any personal opinion. For example, "I watched a movie few days ago."

Subjective classification, on the other hand, is used for sentences that express personal opinions or feelings. These sentences can be further classified as positive or negative. For instance, "It is such a nice movie" would be classified as positive, while "The movie was boring" would be classified as negative.

At the document level, the classification is based on the overall sentiment expressed by the review holder. The document, such as a review, is classified as either positive or negative based on the sentiment conveyed.

In summary, Review Classification can be performed at either the document level or the sentence level. At the sentence level, subjective and objective classifications are carried out, while at the document level, the overall sentiment of the review determines the classification as positive or negative.

Assumption: Each document focuses on a singular object and contains opinions from a sole opinion holder. For instance, thumbs-up or thumbs-down, star ratings (1 star, 2 stars, 3 stars...).

Reviews can also be based on specific features, as exemplified in this example. "This film is incredibly misunderstood, to the point where it's not even amusing. If you're considering watching it for the action scenes, I advise against it. This film delves into the effects and trauma that survivors must endure. Even the detectives are searching for the same answer we all are... WHY? The two leading ladies deliver fantastic performances, showcasing how those we overlook are impacted by the very same things that affect us. Yes, the language can be harsh at times, but it suits the characters perfectly. There are a few loose ends or unanswered questions, but that's common in all movies. The main issues are addressed, and this film makes a significant statement about how adults feel after such major incidents. I highly recommend it for teenagers and adults..."

Each aspect of the product is categorized, and an overall sentiment is evaluated. This project presents a survey on various methods of sentiment analysis found in literature pertaining to product reviews.

### CHAPTER2

1. **LITERATURE SURVEY**

The primary objective of this project is to determine the underlying sentiment of a movie review based on its textual information. Our goal is to classify whether a person enjoyed or disliked the movie based on their review. This is particularly valuable when filmmakers want to assess the overall performance of their movie by analyzing the reviews provided by critics and viewers. The results of this project can also be utilized to create a movie recommender system, which suggests movies to viewers based on their previous reviews. Additionally, this project can help identify groups of viewers with similar movie preferences. As part of this project, we will study various techniques for extracting features from text, such as keyword spotting, lexical affinity, and statistical methods, and analyze their relevance to our problem. Furthermore, we will explore different classification techniques and evaluate their performance with different types of feature representations. Ultimately, we will draw a conclusion regarding the most accurate combination of feature representations and classification techniques for the current predictive task.

The dataset was initially analyzed by researchers at Stanford University, who employed unsupervised learning techniques to group words with similar meanings and generate word vectors. These word vectors were then utilized in various classification models to determine the sentiment polarity of the reviews. This methodology proves particularly valuable when dealing with data that contains extensive sentiment-related content and is susceptible to subjective interpretations of word associations. Additionally, Bo Pang and Peter Turnkey have made significant contributions in the field of polarity detection for both movie and product reviews. Their work also encompasses the development of a multi-class classification system for reviews and the prediction of reviewer ratings for movies and products.

These studies explored the utilization of various classification techniques such as Random Forest classifier, Linear Regression, Decision tree, MLP classifier, and Multinomial naïve Bayes for the purpose of categorizing reviews. Additionally, these works also focused on the implementation of different feature extraction methods. A significant aspect highlighted in these research papers was the exclusion of a neutral category during classification. This decision was based on the assumption that neutral texts tend to be located near the boundary of binary classifiers and pose a disproportionate challenge in terms of classification. Currently, there are numerous sentiment analysis tools and software available either for free or under commercial license. The rise of microblogging has led to the widespread use of sentiment analysis in order to analyze public sentiments and derive meaningful insights from them. One notable example of its application was the utilization of Twitter data to comprehend the political sentiment of individuals in relation to the German Federal elections.

In Section 2.1, we present an overview of how sentiment analysis and classification work in general. In Section 2.2, we discuss some factors that affect sentiment analysis and classification

* 1. **General Overview of Sentiment Analysis and Classification**

The classification of sentiments can be done at different levels, namely the word, sentence, or document level. However, the focus of this particular project is on document-level sentiment classification. Document-level sentiment classification involves determining whether a given set of related documents, which are opinionated in nature, express a positive or negative opinion towards a particular object. In the existing research conducted in this field, it is assumed that the opinionated document (such as a movie review) contains opinions pertaining to a single object. This assumption is generally valid for customer reviews of products and services. However, it may not hold true for forum and blog posts, as these types of posts can contain opinions about multiple products and services.

To determine the sentiment of a document, one possible approach is to analyze the sentiment of each sentence individually and then aggregate the results to obtain the overall sentiment of the document. However, this method requires first classifying the sentiment of each word within the sentences before performing the sentence-level analysis. A major challenge in document-level sentiment analysis is that not all parts of the document contribute equally to inferring the overall sentiment.

Identifying the relevant sentences automatically poses its own learning difficulty. Some researchers have proposed automated methods for extracting meaningful sentences, while others have explored algorithms that consider all sentences in the document to achieve improved experimental outcomes.

* 1. **Factors that affect sentiment analysis and classification**

classifying a document as positive or negative based on the overall sentiment expressed by the author is considered a more challenging task compared to strict text classification. This is because opinions can often be conveyed in a more intricate manner, making it difficult to identify them solely based on individual words within a sentence or document. Merely analyzing the words used in a review may not provide an accurate classification of the sentiments expressed. Take, for instance, this review: "This film should be brilliant. It sounds like a great plot, the actors are first

grade, and the supporting cast is good as well, and Stallone is attempting to deliver a good performance, however, it can't hold up".

The utilization of terms like "brilliant," "great," and "good" implies a favorable sentiment, leading one to believe that determining the sentiment of a review through a predefined set of keywords would be straightforward. Nevertheless, Pang and Lee's experimental findings revealed that the accuracy of classification achieved through human-generated keyword lists was inferior to that of keywords generated through machine learning techniques.

The review mentioned above contains an anaphor, such as the phrase "it can't hold up." The referent of "it" in this phrase is unclear, whether it pertains to the movie or Stallone's performance. Consequently, it becomes challenging to ascertain the overall sentiment of the review. Numerous free-form reviews employ multiple anaphora, abbreviations, and exhibit a lack of capitalization, improper spelling, punctuation, and grammar.

The analysis and classification of opinions are influenced by several key factors. These factors include the domain of the datasets, the size of the datasets being analyzed, the format of the datasets (whether they are labeled or unlabeled), and the quality of the dataset. The accuracy of sentiment classification can be affected by the specific domain to which it is applied. This is because the same phrase can have different sentiments in different domains. For example, the phrase "go read the book" is likely to indicate positive sentiment in book reviews, but negative sentiment in movie reviews. Similarly, a review like "it is so easy to predict the next action..." may be negative sentiment for a movie plot, but positive sentiment for a political review. Additionally, the variation in vocabularies across different domains poses a challenge when applying classifiers trained on labeled data from one domain to test data from another domain.

Depending on the size of datasets to be utilized, either manual or automatic, or both methodologies, could be employed. Nevertheless, it is always advisable to employ a combination of both approaches. Numerous experiments have revealed that even the poorest outcomes obtained from utilizing both approaches surpass the best outcomes achieved through manual approaches and certain automatic approaches. The presence of labeled data also enhances the efficiency of opinion classification. In its

absence, many researchers have resorted to the linguistic/semantic approach of constructing lexicons, which is exceedingly time-consuming and yet does not yield superior performance. Furthermore, lexicons are language and domain-specific, thereby further complicating the sentiment analysis and classification task.

The sentiment classification performance is directly influenced by the quality of the dataset. As there is currently no mechanism in place to ensure the quality of reviews, anyone can post anything on the internet. Consequently, a significant number of low-quality reviews and review spam exist. Bing et al. conducted a study on opinion spam and found that online reviews often consist of spam messages, including false or fabricated reviews, irrelevant reviews, and reviews that are not genuine but rather statements or questions. To mitigate the presence of spam in these reviews, it is crucial to implement effective pre-processing techniques. However, it is worth noting that this particular research area has not received much attention thus far.

### CHAPTER3

**SYSTEM ANALYSIS**

##### EXISTING SYSTEM

The current process of categorizing reviews is carried out manually by reading each review and assigning it a positive or negative label. However, this system is unable to handle large volumes of data within the given timeframe. Moreover, this approach is limited to processing structured data and the accuracy of existing technologies is incomparable to that of modern technologies.

##### PROBLEM STATEMENT

The current system experiences a longer duration for review classification due to manual processing. Consequently, it becomes exceedingly challenging to generate results for the vast amount of data that exists in today's world.

##### PROPOSED SYSTEM

The classification of Movie Reviews into specific categories can be addressed and simplified by employing supervised machine learning algorithms for text classification. This project aims to develop a machine learning model that automates the process of categorizing Movie Reviews into either positive or negative sentiments. Various models such as Random Forest, Decision Tree, MLP, Logistic Regression, etc. will be trained using IMDB data obtained from the official IMDB website. The model with the highest accuracy will be deemed the most suitable for this project.

### FEASIBILITY STUDY

In this phase, the project's feasibility is examined and a business proposal is presented, outlining a broad plan for the project along with cost estimates. The proposed system undergoes a thorough feasibility study during the system analysis stage. This is done to ensure that the implementation of the proposed system does not impose any unnecessary burden on the company. It is crucial to have a clear understanding of the key requirements for the system in order to conduct a comprehensive feasibility analysis.

Three key considerations involved in the feasibility analysis are

##### ECONOMICAL FEASIBILITY

1. **TECHNICAL FEASIBILITY**

##### SOCIAL FEASIBILITY

* + 1. **ECONOMICAL FEASIBILITY**

This investigation aims to assess the financial consequences that the system will have on the organization. The organization's capacity to allocate funds for the research and development of the system is restricted. Therefore, the expenses must be rationalized. Consequently, the system was developed within the allocated budget, primarily utilizing freely available technologies. The only exception was the acquisition of customized products.

### TECHNICAL FEASIBILITY

This research is conducted to assess the technical viability, specifically the technical prerequisites of the system. It is crucial that any system created does not impose excessive strain on the existing technical resources. Otherwise, it will result in an increased burden on the client. Therefore, the developed system should have a moderate requirement, necessitating minimal or no alterations for its implementation.

### SOCIAL FEASIBILITY

The purpose of studying is to assess the degree of acceptance of the system by the user. This encompasses the user training process to ensure efficient utilization of the system.

##### Software Requirement Specification

* + 1. **Software Requirements**

Operating System : Windows, Linux or UNIX Software : Anaconda

IDE : Jupyter notebook Editor : Text Editor Toolkit : NLTK

Code Language : Python 3.6

##### Hardware Requirements

Processor : Intel i3, i5, i7 and above Hard Disk : 100 GB

RAM : 4 GB or above

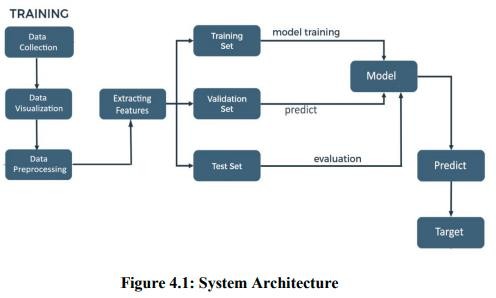
## CHAPTER 4

**SYSTEM DESIGN**

## 4.1 SYSTEM ARCHITECTURE

In this section, we present a concise outline of the sentiment analysis and classification process. For a more comprehensive explanation, please refer to Chapter 5. Figure 1.1 illustrates the key stages involved in text classification using sentiment. The initial step, known as pre-processing, entails extracting the reviews or documents from a source dataset. Subsequently, various techniques are employed to clean the terms within each review. These terms, along with their corresponding sentiment scores, are then stored as feature vectors, which serve as input for a text classifier.

The subsequent phase entails employing a text classifier to categorize the chosen attributes as either positive or negative. The feature vectors that have been stored are utilized as input for the text classifier. If deemed necessary, feature selection can be implemented to diminish the quantity of attributes. To achieve optimal outcomes, a sequence of iterative measures, referred to as cross-validation, can be employed to assess the predictive model's practical performance accurately.



### Data Collection

The AI department of Stanford University utilized the Large Movie Review Dataset to gather the dataset for this task. This dataset, consisting of 50,000 training examples, was collected from IMDb. Each review in the dataset is labeled with the movie's rating on a scale of 1-10.

##### Data Visualisation

Data visualization refers to the visual depiction of data, where images are created to effectively convey the relationships within the data to the viewers. This is accomplished by employing a structured mapping between graphic elements and data values during the creation of the visualization. This mapping determines how data values will be visually represented, dictating the extent to which properties of the graphic elements, such as size or color, will alter to reflect changes in the data values.

##### Data Pre-processing

Data pre-processing plays a crucial role in the data mining process. The concept of "garbage in, garbage out" holds true in the realm of data mining and machine learning endeavors. Often, data collection methods lack strict control, leading to the inclusion of out-of-range values (e.g., Income: −100), implausible data combinations (e.g., Sex: Male, Pregnant: Yes), missing values, and more. Analyzing data without carefully addressing these issues can result in misleading outcomes. Therefore, ensuring the accuracy and quality of data is paramount before conducting any analysis. In many cases, data pre- processing serves as the most vital phase of a machine learning project, particularly in the field of computational biology.

**Extracting Features**

Text processing involves dealing with words in a text as discrete and categorical features. To make this data usable for algorithms, we need to encode it in a specific manner. This process of transforming textual data into real-valued vectors is known as feature extraction. Two common techniques for numerical representation of text are Bag of Words and TF-IDF.

# Data Splitting

Training Dataset: The sample of data used to fit the model.

Test Dataset: The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.

Validation Dataset: The data sample is utilized to impartially assess the fitting of a model on the training dataset during the process of tuning the model's hyperparameters. However, as the model configuration incorporates the skill on the validation dataset, the evaluation becomes increasingly biased.

Model Building

The overall task in this project is for classification of reviews as favourable or

unfavourable. Therefore, for this classification task we explored multiple classification models on above feature representations We employed a variety of models, starting from the basic Logistic Regression to the advanced SVM Classifier. Additionally, we utilized other classification models such as MLP Classifier, Decision Tree, and Random Forest Classifier. In addition to these, we trained the aforementioned feature representations on Naïve Bayes' Classifier, which is commonly used in text mining alongside Bag of Words and TF-IDF Modelling. Furthermore, we developed a model based on k-Nearest Neighbours to assess the similarity between reviews and classify them accordingly.

Predicting Target

The feature of a dataset that you aim to comprehend better is referred to as the target variable. In supervised machine learning, historical data is utilized to identify patterns and establish connections between the target variable and other features in the dataset. The specific target variable will differ based on the business objective and the data that is accessible.

## CHAPTER 5

##### SYSTEM IMPLEMENTATION

**Dataset**

The AI department of Stanford University utilized the Large Movie Review Dataset for this task. This dataset consists of 50,000 training examples gathered from IMDb, where each review is labeled with a movie rating ranging from 1 to 10. Since sentiments are often binary, such as good/bad or happy/sad, we categorized these ratings as either 1 (like) or 0 (dislike) based on the ratings. If the rating was above 5, we inferred that the individual liked the movie; otherwise, they did not. Initially, the dataset was split into two subsets, each containing 25,000 examples for training and testing. However, we found this division to be sub-optimal due to the limited number of training examples, resulting in under-fitting. To address this, we redistributed the examples, allocating 40,000 for training and 10,000 for testing. Although this yielded better models, it also led to over-fitting on the training examples and poorer performance on the test set. Ultimately, we decided to employ Cross Validation, dividing the complete dataset into multiple folds with different samples for training and validation in each iteration. The final performance statistic of the classifier is then averaged over all results, resulting in improved accuracy across the board. A typical review text within the dataset appears as follows:.

I am a fan of television movies in general, and this particular one was quite impressive. The performances by the cast were consistently strong, and there were unexpected twists before each commercial break. It reminded me of a combination of "Medium" and "CSI."

Did anyone else notice that, under certain lighting, the daughter resembled a young Nicole Kidman? Is there any relation between them? I would definitely watch it again or rent it if it becomes available on video.

Dedee was fantastic in her role. I haven't seen her in many projects, but she portrayed her character convincingly. If you enjoy TV mystery movies, I recommend checking this one out if you get the chance.

In the text provided, we had to remove HTML tags like "<br>" as a pre-processing step before extracting features. We used simple regular expressions to accomplish this.

Additionally, we made the text case-insensitive to facilitate counting word occurrences across all reviews and removed punctuation marks such as '!', '?' and others, as they do not contribute significant information and can have different connotations. These tasks were performed using standard Python libraries for text and string manipulation. Furthermore, we eliminated stop words from the text for certain feature extraction tasks, which will be discussed in more detail later on. It is important to note that we did not employ word stemming, as it can result in the loss of information by reducing words to their root forms.

##### Data Pre-processing

Prior to executing any algorithm, it is imperative to ensure the data is thoroughly cleansed, facilitating smoother processing. Furthermore, by proactively identifying and eliminating extraneous words, we can significantly enhance the precision of our algorithms.

**Cleaning the data**

The presence of HTML tags in the IMDb reviews does not contribute to sentiment detection. Therefore, we have made the decision to eliminate all punctuation, including emoticons (which are already scarce). This simplifies our data processing. Additionally, we convert all text to lowercase. Furthermore, we utilize the Porter stemming algorithm to replace each word with its root form. As a result, words such as "cats" and "cat," or "running" and "run," are treated as identical. This approach has been proven to enhance the accuracy of sentiment analysis in classification tasks.

##### Stemming and lemmatization

Different forms of a word, such as organize, organizes, and organizing, are utilized in documents for grammatical purposes. Moreover, there exist families of derivationally related words with comparable meanings, like democracy, democratic, and democratization. In numerous instances, it appears advantageous for a search to yield documents containing any word from the aforementioned set.

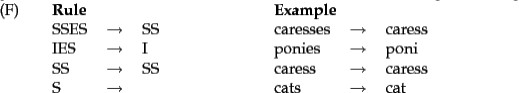
Both stemming and lemmatization aim to reduce the various forms of a word, including inflectional and derivationally related forms, to a shared base form. For example:

am,are,is => be

car, cars, car's, cars => car

The result of this mapping of text will be something like:

The boy's cars are differentcolors => the boy car be differ color

Porter's algorithm is widely recognized as the most commonly used and highly effective algorithm for stemming English words. Although the complete algorithm is complex and extensive, we will provide a brief overview of its general approach. The algorithm comprises five distinct phases of word reductions, which are applied sequentially. Each phase incorporates specific conventions for selecting rules, such as choosing the rule from each rule group that is applicable to the longest suffix. In the initial phase, this convention is employed along with the following rule group:

##### Tokenization

Tokenization is the process of dividing a character sequence into smaller units, known as tokens, while also discarding specific characters like punctuation marks. Let's consider an illustration of tokenization:

Input: Friends, Romans, Countrymen, lend me your ears;.

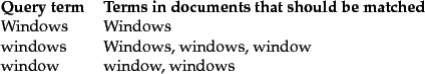
Output:



## Normalization

Token normalization involves the normalization of tokens to ensure that matches occur even when there are slight differences in the character sequences of the tokens. The common approach to normalization is to create equivalence classes, typically named after one member of the set. For example, if the tokens "anti- discriminatory" and "antidiscriminatory" are both mapped to the term "antidiscriminatory" in both the document text and queries, searching for either term will retrieve documents that contain it.

Using mapping rules that eliminate characters such as hyphens offers the advantage of implicit equivalence classing. This means that the calculation of equivalence classes is not required beforehand, as the terms that become identical through these rules automatically form the equivalence classes. It is relatively simple to create rules that remove characters. However, since the equivalence classes are implicit, it is not always clear when it is necessary to add characters. For example, it would be challenging to determine that "antidiscriminatory" should be transformed into "anti-discriminatory".



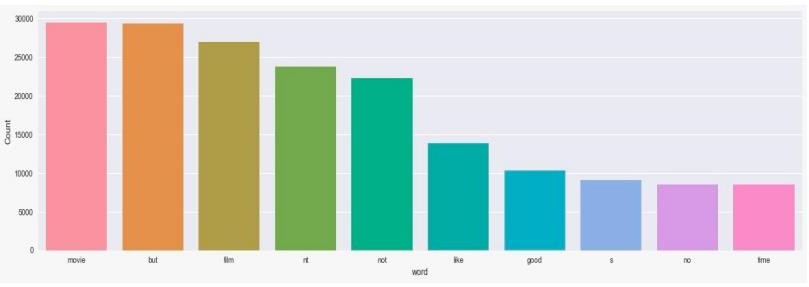
Instead of creating equivalence classes, an alternative approach is to establish relationships between unnormalized tokens. This technique can also be applied to curated lists of synonyms like car and automobile. These relationships between terms can be achieved in two different ways. The common approach involves indexing unnormalized tokens and maintaining a query expansion list that includes multiple vocabulary entries to consider for a specific query term. As a result, a query term becomes a disjunction of several postings lists. The alternative approach involves expanding the index during its construction. For instance, when a document contains the term "automobile," it is indexed under "car" as well (and vice versa in most cases). However, both of these methods are less efficient compared to equivalence classing since they require storing and merging more postings. The first method adds a query expansion dictionary and requires additional processing during query time, while the second method necessitates more storage space for the postings. Traditionally, expanding the space required for postings lists was considered a disadvantage. However, with the decreasing costs of modern storage, the increased flexibility provided by distinct postings lists has become more appealing.

The flexibility of these methods surpasses that of equivalence classes due to the possibility of overlapping expansion lists without being identical. Consequently, an asymmetry in expansion can occur. Figure 4 illustrates an example of how this asymmetry can be utilized. For instance, if the user inputs "windows," we aim to enable matches with the capitalized "Windows" operating system. However, if the user enters "window," it is not feasible to consider it a match, despite the possibility of it also matching lowercase "windows."

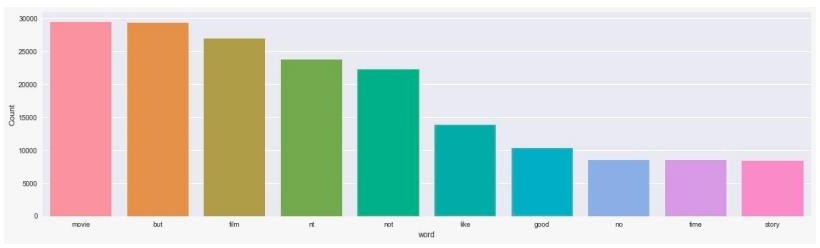
##### Stop Words



Occasionally, certain commonly used words that may seem insignificant in assisting with document selection for a user's needs are completely excluded from the vocabulary. These words are referred to as stop words. The typical approach for determining a stop list is to arrange the terms based on their frequency in the document collection and then select the most frequent terms, often filtered manually for their semantic relevance to the domain of the indexed documents. These selected terms form the stop list, and they are disregarded during the indexing process. An example of a stop list is displayed in the above Figure. By utilizing a stop list, the number of postings that a system needs to store is significantly reduced. Omitting the indexing of stop words often has minimal impact on keyword searches that include terms like "the" and "by," as they do not appear to be very useful. However, this is not the case for phrase searches. For instance, the phrase query "President of the United States," which contains two stop words, is more precise than using "President" AND "United States." If the word "to" is included in the stop list, the meaning of searches for flights to London may be lost. Similarly, searching for Vannevar Bush's article "As we may think" becomes challenging if the first three words are excluded, and the system only searches for documents containing the word "think." Certain types of queries are disproportionately affected by stop words. Some song titles and well-known verses consist entirely of words that commonly appear on stop lists (e.g., "To be or not to be," "Let It Be," "I don't want to be..."). The trend in information retrieval (IR) systems has shifted from using relatively large stop lists (200-300 terms) to very small stop lists (7-12 terms) and, in some cases, eliminating stop lists altogether. Web search engines typically do not employ stop lists. The design of modern IR systems has focused on leveraging language statistics to better handle common words.



**Figure 5.3: Before removing stop words**



**Figure 5.4 After removing stop words**

#### Data Visualisation

Data visualization refers to the visual depiction of data, where images are created to effectively convey the relationships within the data to the viewers. This is accomplished by employing a structured mapping between graphic elements and data values during the creation of the visualization. This mapping determines how data values will be visually represented, dictating the extent to which properties of the graphic elements, such as size or color, will alter to reflect changes in the data values.

#### Tag Clouds

Tag clouds are the most basic and widely used method of visualizing text. They display tags in a spatial arrangement, with variations in size, color, and position reflecting the frequency, categorization, or importance of each tag.



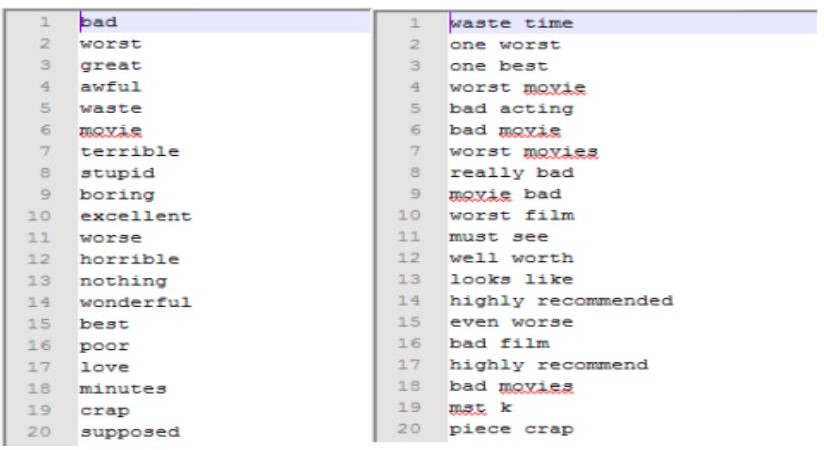
#### Figure 5.5: Word Cloud

**Extracting Features**

Text processing involves dealing with words in a text as discrete and categorical features. To make this data usable for algorithms, we need to encode it in a specific manner. This process of transforming textual data into real-valued vectors is known as feature extraction. Two common techniques for numerical representation of text are TF-IDF and Bag of Words.

**TF-IDF**

Tf-Idf, short for term frequency, inverse document frequency, is a valuable technique primarily employed in information retrieval to assess the significance of a keyword in a specific document within a corpus1. Reflecting on the term "movie," it led us to realize that there was still irrelevant content, prompting us to consider the possibility that a word could hold great importance in the entire collection of reviews but not in any individual review. Consequently, we required a method to determine the importance of words in relation to each review.



#### Figure 5.6: Tf-Idf

**Bag of Words**

The bag-of-words model is a method used in natural language processing to represent text. Instead of considering grammar or linguistic structures, this model treats a text as a collection of its individual words. After preprocessing the data, let's assume there are N unique words in the entire document. Additionally, suppose the document contains R reviews. Each review is represented by an N-dimensional feature vector. The entries in these vectors correspond to the N words in the document, and the feature values indicate the frequency of each word in the review. Alternatively, we can create a binary feature vector where 1 represents the presence of a word in the review, and 0 represents its absence. However, to handle the large size of the text document, we need to perform "feature selection" to reduce the dimensionality of the data points. After preprocessing and cleaning the data, we have a total of N = 78,767 words in the document. These words are ranked based on their importance, determined using the mutual information criteria discussed earlier. We select the top d = 5000 words to construct the feature vectors. It's worth noting that a larger number of words would likely yield more accurate results, but we are constrained by memory limitations and algorithm runtime.

#### Data Splitting

Once the reviews have been cleaned and pre-processed, the data can be divided into three sets: train, validation, and test. These sets will be split in the ratio of 70%, 15%, and 15% respectively. The train set will be utilized for training the model, while the validation and test sets will be employed for making predictions and evaluating the model.

**Training Dataset:** The sample of data used to fit the model.

**Test Dataset:** The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.

**Validation Dataset:**The data sample is utilized to impartially assess the fitting of a model on the training dataset during the process of tuning the model's hyperparameters. However, as the model configuration incorporates the skill on the validation dataset, the evaluation tends to become more biased.

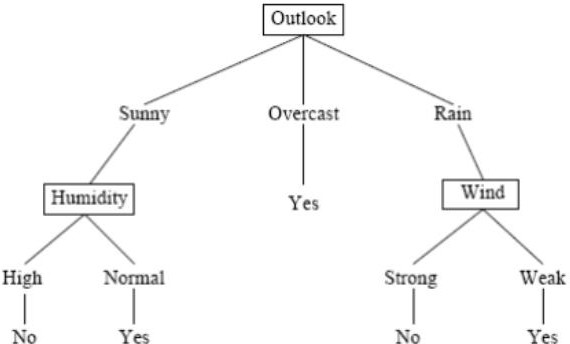
#### Model Building

The primary objective of this project is to classify reviews as either favorable or unfavorable. To accomplish this classification task, we explored various classification models using the aforementioned feature representations. Our models ranged from the simple Logistic Regression to the state-of-the-art SVM Classifier. Additionally, we utilized other classification models such as MLP Classifier, Decision Tree, and Random Forest Classifier. In addition to these models, we also trained the above feature representations on Naïve Bayes' Classifier, which is commonly used in text mining in conjunction with Bag of Words and TF-IDF Modelling. Furthermore, we developed a model based on k-Nearest Neighbors to determine the similarity between reviews and classify them accordingly.

#### Decision Tree Classifier

A decision tree is a hierarchical structure where the internal nodes represent tests based on attributes, and the leaf nodes represent categories or classes. Each internal node tests a specific attribute, and each branch from a node corresponds to a value for that attribute.

The decision-making attribute is not predetermined, allowing for the selection of the attribute that provides the most information. Decision trees are not restricted to Boolean functions and can be applied to functions with categorical values as well.



#### Figure 5.9: Decision Tree

In the given example, the provided instances can be categorized by the values of the "outlook" attribute. The instances are divided based on attributes, and the decision for each node is determined by selecting the attribute that provides the most information. Therefore, in this example, choosing "Outlook" as the root node yields the highest amount of information at that level. The edges represent the possible values of the attributes, and the instances are divided accordingly among the child nodes. The tree can be a multi-way tree depending on the potential attribute values. The selection of attributes is based on a heuristic approach, where the chosen attribute is expected to provide the best division at a specific level. This approach has proven to be successful in the past.

#### Random Forest Classifier

A Random Forest is a type of classifier that is composed of multiple tree-structured classifiers. Each tree in the forest independently generates random vectors and casts a vote for the most popular class at a given input x. The random vectors are generated independently of each other, with the same distribution, and a tree is created using the training set. In the case of random forests, a upper bound is derived to estimate the generalization error based on two parameters: the accuracy of the individual classifiers and the dependency between them.

The error generalization for random forest can be divided into two components. These components are described as follows:

1. The effectiveness of each individual classifier within the forest.
2. The level of correlation between the classifiers based on the raw margin function.

To enhance the precision of the random forest, it is essential to reduce the correlation while preserving its robustness. According to Brieman's (2001) research, forests are constructed by randomly selecting inputs or combinations of inputs at each node to grow the tree. This class of procedures possesses several advantageous qualities, which are outlined as follows:

1. Demonstrates good accuracy, and in some cases, even superior performance.
2. Exhibits relative resilience to outliers and noise.
3. Operates at a faster pace compared to Bagging and Boosting techniques.
4. Possesses simplicity and can be easily parallelized.

Brieman has proposed a randomization approach that is more effective when used with bagging or the random space method. The process of generating each tree in a random forest involves the following steps:

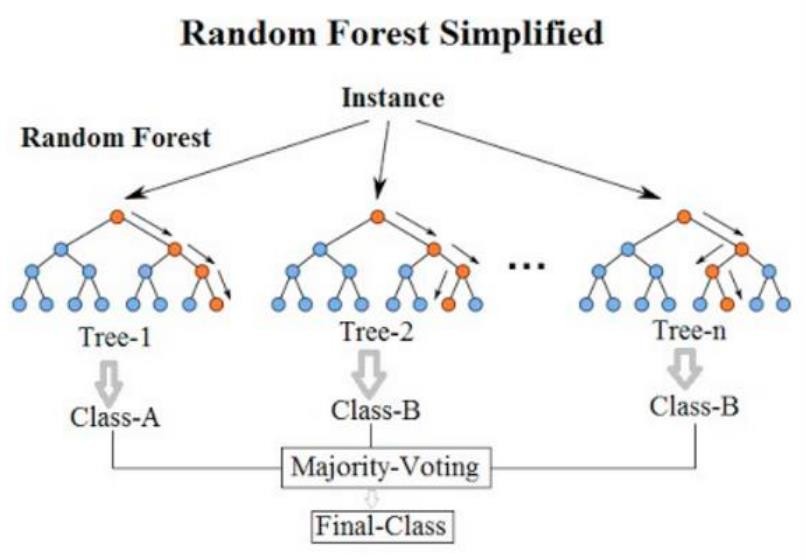
1. The training dataset contains N records.
2. N records are randomly sampled with replacement.
3. This sampled dataset is referred to as the bootstrap sample.
4. If the training set consists of M input variables, a random selection of m<<M inputs is made, and the best split on these m attributes is used to divide the node.
5. The value of m remains constant throughout the growth of the forest.
6. The tree is grown to its maximum possible level.

There are a couple of justifications for employing the Bagging approach, as outlined by Brieman (1994). Firstly, it appears that incorporating bagging in conjunction with random features yields more precise outcomes. Secondly, bagging can be utilized to continuously estimate generalization error, as well as the strength and correlation.

When a new instance needs to be classified, the forest undergoes a process where each tree in the forest provides a vote for the classification of the instance. These votes are then combined and counted, and the classification with the highest number of votes is declared as the classification for the new instance. This process is referred to as the Forest RI process.

To build the forest, a bootstrap sample is created by randomly selecting data with replacement for each tree. As a result, one-third of the instances are left out and referred to as Out of Bag (OOB) data. Each tree in the forest has its own OOB data, which is used to estimate the error of individual trees, known as OOB error estimation.

In addition, random forests also include built-in features for calculating variable importance and proximities.



**Figure 5.10: Random Forest**

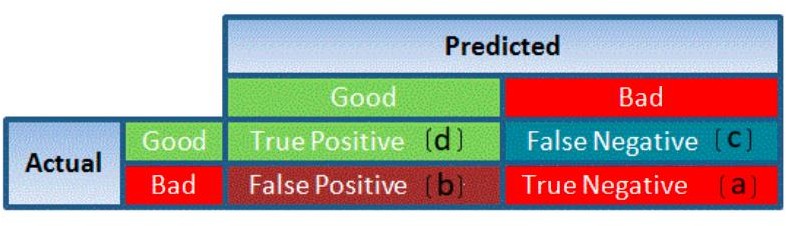
## CHAPTER 6

#### SYSTEM TESTING

**Model Evaluation**

The evaluation of a model's performance is an essential step in the model building process. In order to determine the accuracy of predictions, it is necessary to compare them with the actual values. This can be done by plotting the results and calculating the distance between the predictions and actual values. The accuracy of the predictions increases as the distance between them decreases. As this is a classification problem, there are several evaluation metrics that can be used to assess the performance of our models.

* **Precision**: We can gain a better understanding of accuracy by referring to the confusion matrix, which presents a tabular format comparing the actual and predicted values. The confusion matrix provides a visual representation of the data, allowing us to assess the accuracy of our predictions.



#### Figure 6.1 : Accuracy

1. True Positive refers to targets that are genuinely true (Y) and have been accurately predicted as true (Y).

True Negative refers to targets that are genuinely false (N) and have been accurately predicted as false (N).

False Positive refers to targets that are genuinely false (N) but have been incorrectly predicted as true (T).

False Negative refers to targets that are genuinely true (T) but have been incorrectly predicted as false (N).

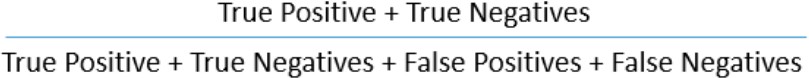
1. True Positive denotes targets that are indeed true (Y) and have been correctly predicted as such (Y).

True Negative denotes targets that are indeed false (N) and have been correctly predicted as false (N).

False Positive denotes targets that are indeed false (N) but have been erroneously predicted as true (T).

False Negative denotes targets that are indeed true (T) but have been erroneously predicted as false (N).

Using these values, we can calculate the accuracy of the model. The accuracy is given by:



* + **Precision**: refers to the level of accuracy attained in accurate predictions, specifically in terms of correctly labeled observations. It quantifies the proportion of true observations that are correctly labeled as true. The formula to calculate precision is TP / (TP + FP).
  + **Recall**, also referred to as Sensitivity, is a metric that quantifies the number of correctly predicted observations of the true class. It is calculated by dividing the

number of true positives (TP) by the sum of true positives and false negatives (TP

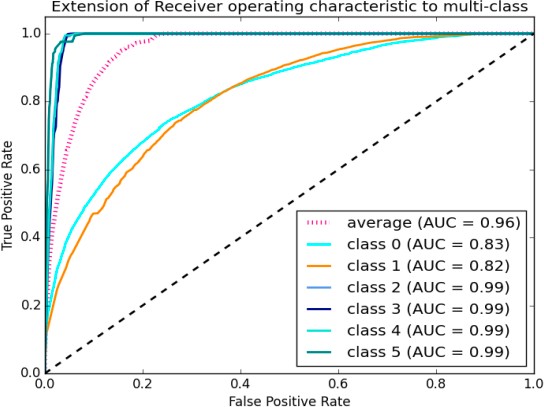
+ FN).

* + **Specificity** refers to the accuracy of correctly identifying observations of the false class. It is calculated by dividing the true negatives (TN) by the sum of true negatives and false positives (TN + FP).

Specificity and Sensitivity plays a crucial role in deriving ROC curve.

* The ROC curve is a graphical representation that summarizes the performance of a model by analyzing the balance between the true positive rate (sensitivity) and the false positive rate (1-specificity).
* The area under the curve (AUC), also known as the index of accuracy (A) or concordance index, serves as an excellent performance metric for the ROC curve. A higher AUC indicates a stronger predictive power of the model.

This is how a ROC curve looks like:



#### Figure 6.2 : ROC curve

The curve's area quantifies the model's ability to accurately classify true positives and true negatives. Our objective is for the model to correctly predict true classes as true and false classes as false.

Hence, it can be stated that we aim for a true positive rate of 1. However, we are not solely concerned with the true positive rate, but also with the false positive rate. In our specific problem, we not only want to predict the Y classes as Y, but also want the N classes to be predicted as N.

We strive to maximize the area under the curve, which is highest for class 2, 3, 4, and 5 in the aforementioned example.

For class 1, when the false positive rate is 0.2, the true positive rate is approximately 0.6. Conversely, for class 2, the true positive rate is 1 at the same false positive rate.

Consequently, the AUC for class 2 will be significantly higher compared to the AUC for class 1. Therefore, the model for class 2 will be superior.

The models for class 2, 3, 4, and 5 will predict with greater accuracy compared to the models for class 0 and 1, as the AUC is higher for those classes.

#### RESULT ANALYSIS

This project uses above mentioned models with both BOW and TF-IDF. The accuracies of the models are compared .The model which gets best accuracy is taken.

#### CONCLUSION

Review classification using NLTK, Spacy, Scikit-learn is a machine learning model which uses modern Natural language processing techniques. With this Project, the people can easily classify the review of the movie without putting much effort .The modern algorithms in this project plays key role in getting high accuracy for a given data. The MLP classifier gives the High accuracy for this project. Hence we can conclude that MLP classifier is the best fitting model for review Classification.

#### FUTURE ENHANCEMENTS

Text classification involves assigning relevant categories from a predefined set to natural language texts. In simpler terms, it is the extraction of generic tags from unstructured text. These generic tags are derived from a predetermined set of categories.

## REFERENCES

1. Liu, B., "Investigating User Opinions in Recommender Systems", Proceedings of the Second KDD Workshop on Large Scale Recommender Systems and the Netflix Prize Competition, August 24, 2008, Las Vegas, Nevada, USA.
2. Dave, D., Lawrence, A., Pennock, D., "Analyzing the Peanut Gallery: Extracting Opinions and Classifying Product Reviews", Proceedings of the International World Wide Web Conference (WWW‟03), 2003.
3. Turney, P., "Thumbs Up or Thumbs Down? Applying Semantic Orientation to Unsupervised Review Classification", ACL‟02, 2002.
4. Porter, M., "A Stemming Algorithm for Text Classification", Retrieved from <http://people.scs.carleton.ca/~armyunis/projects/KAPI/porter.pdf>.
5. An Introduction to Logistic Regression Analysis and Reporting, Article in The Journal of Educational Research, September 2002.
6. Mitchell, T., "Decision Trees for Text Classification", Retrieved from <https://www.cs.princeton.edu/courses/archive/spr07/cos424/papers/mitchell-dectrees.pdf>.
7. Xu, B., "An Enhanced Random Forest Classifier for Text Categorization", Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China.
8. Knigam, C., "Multinomial Logistic Regression for Text Classification", Retrieved from <http://www.cs.cmu.edu/~knigam/papers/multinomial-aaaiws98.pdf>.
9. Suykens, J. A. K., Vandewalle, J., "Training Multilayer Perceptron Classifiers Using a Modified Support Vector Method".