SENTIMENT ANALYSIS ON IMDB REVIEWS USING MACHINE LEARNING ALGORITHMS

Submitted in partial fulfillment of requirements for award of the degree of

Bachelor of Technology
In
Information Technology

Guided By:
Dr. Anuradha Chugh
Assistant Professor
USICT

Submitted by:
Satyam Sehgal
00916401513
B.Tech IT (8th Sem)

University School of Information and Communication Technology
Guru Gobind Singh Indraprastha University
(2013-2017)

Certificate

This is to certify that the Seminar & Progress Report (IT 452) entitled “Sentiment Analysis on IMDB reviews using Machine Learning Algorithms” done by Mr. Satyam Sehgal, Roll No. 00916401513 is an authentic work carried out by him at USICT under my guidance. The matter embodied in this Seminar & Progress Report has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief.

Date:

Satyam Sehgal
00916401513

Dr. Anuradha Chugh

<u>ACKNOWLEDGEMENT</u>

I am using this opportunity to express my gratitude to everyone who supported me throughout the course of this project. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

I express my warm thanks to **Dr. Anuradha Chugh** for her support and guidance and also I would like to thank all the people who provided me with the facilities being required and conductive conditions for my project.

Thank you,

Satyam Sehgal

ABSTRACT

Learning good semantic vector representations for phrases, sentences and paragraphs is a challenging and ongoing area of research in natural language processing and understanding. In this project, we have taken a Large Movie Review Dataset of IMDB and based on the reviews given by the user, the reviews are classified into a positive review and a negative review. The first task in this project is to convert movie reviews or words into vector points or data points and then classification is applied on these data points which predicts the given movie review as positive or negative. For classification we are using four approaches: SVM (Support Vector Machine), Naïve-Bayes, Random Forest and Bagged Tree approach. These approaches classify the given data points but they differ to a huge extent in their accuracy.

When the data points have been extracted from the words using NLP (Natural Language Processing), then this data is used to train the machine so that the machine can predict the output of the other data which has to be tested which is also known as test data. Thus the result of the test data will be the output of the machine

Contents

<u>List of figures</u>

Chapter 1

1.1 MACHINE LEARNING

Machine learning is the subfield of computer science that, according to Arthur Samuel in 1959, gives "computers the ability to learn without being explicitly programmed." Evolved from the study of pattern recognition and computational learning theory in artificial intelligence, machine learning explores the study and construction of algorithms that can learn from and make predictions on data – such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions, through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or unfeasible; example applications include email filtering, detection of network intruders or malicious insiders working towards a data breach, optical character recognition (OCR), learning to rank and computer vision.

Machine learning is the science of getting computers to act without being explicitly programmed. In the past decade, machine learning has given us self-driving cars, practical speech recognition, effective web search, and a vastly improved understanding of the human genome. Machine learning is so pervasive today that we probably use it dozens of times a day without knowing it. Many researchers also think it is the best way to make progress towards human-level AI.

As a scientific endeavor, machine learning grew out of the quest for artificial intelligence. Already in the early days of AI as an academic discipline, some researchers were interested in having machines learn from data. They attempted to approach the problem with various symbolic methods, as well as what were then termed "neural networks"; these were mostly perceptions and other models that were later found to be reinventions of the generalized linear models of statistics. Probabilistic reasoning was also employed, especially in automated medical diagnosis.

However, an increasing emphasis on the logical, knowledge-based approach caused a rift between AI and machine learning. Probabilistic systems were plagued by theoretical and practical problems of data acquisition and representation. By 1980, expert systems had come to dominate AI, and statistics was out of favor. Work on symbolic/knowledge-based learning did continue within AI, leading to inductive logic programming, but the more statistical line of research was now outside the field of AI proper, in pattern recognition and information retrieval. Neural networks research had been abandoned by AI and computer science around the same time. This line, too, was continued outside the AI/CS field, as "connectionism", by researchers from other disciplines including Hopfield, Rumelhart and Hinton. Their main success came in the mid-1980s with the reinvention of back propagation.

Machine learning, reorganized as a separate field, started to flourish in the 1990s. The field changed its goal from achieving artificial intelligence to tackling solvable problems of a practical nature. It shifted focus away from the symbolic approaches it had inherited from AI, and toward methods and models borrowed from statistics and probability theory. It also benefited from the increasing availability of digitized information, and the possibility to distribute that via the Internet.

Machine learning is closely related to (and often overlaps with) computational statistics, which also focuses on prediction-making through the use of computers. It has strong ties to mathematical optimization, which delivers methods, theory and application domains to the field. Machine learning is sometimes conflated with data mining, where the latter subfield focuses more on exploratory data analysis and is known as unsupervised learning. Machine learning can also be unsupervised and be used to learn and establish baseline behavioral profiles for various entities and then used to find meaningful anomalies.

Machine learning is a type of artificial intelligence that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can change when exposed to new data. The process of machine learning is similar to that of data mining. Both systems search through data to look for patterns. However, instead of extracting data for human comprehension -- as is the case in data mining applications -- machine learning uses that data to detect patterns in data and adjust program actions accordingly. Machine learning algorithms are often categorized as being supervised or unsupervised. Supervised algorithms can apply what has been learned in the past to new data. Unsupervised algorithms can draw inferences from datasets.

Machine learning tasks are typically classified into three broad categories, depending on the nature of the learning "signal" or "feedback" available to a learning system. These are:

- **Supervised learning**: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.

- **Unsupervised learning**: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).

- **Reinforcement learning**: A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). The program is provided feedback in terms of rewards and punishments as it navigates its problem space.

Between supervised and unsupervised learning is semi-supervised learning, where the teacher gives an incomplete training signal: a training set with some (often many) of the target outputs missing. Transduction is a special case of this principle where the entire set of problem instances is known at learning time, except that parts of the targets are missing.

Among other categories of machine learning problems, learning to learn learns its own inductive bias based on previous experience. Developmental learning, elaborated for robot learning, generates its own sequences (also called curriculum) of learning situations to cumulatively acquire repertoires of novel skills through autonomous self-exploration and social interaction with human teachers and using guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

Another categorization of machine learning tasks arises when one considers the desired output of a machine-learned system:

- In classification, inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one or more (multi-label classification) of these classes. This is typically tackled in a supervised way. Spam filtering is an example of classification, where the inputs are email (or other) messages and the classes are "spam" and "not spam".
- In regression, also a supervised problem, the outputs are continuous rather than discrete.
- In clustering, a set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.
- Density estimation finds the distribution of inputs in some space.
- Dimensionality reduction simplifies inputs by mapping them into a lower-dimensional space. Topic modeling is a related problem, where a program is given a list of human language documents and is tasked to find out which documents cover similar topics.

1.2 <u>SENTIMENT ANALYSIS</u>

Sentiment Analysis also termed as opinion mining is the mining of opinions of individuals, their appraisals, and feelings in the direction of certain objects, facts and their attributes. As stated by Liu and Jawale in the latest years, opinion mining has attracted great deal of concentration from both the academicians and industry persons because of various challenging research issues and support of sentiment analysis for a broad set of applications. Opinions play a very important role in making a proper decision. As it is wise to get or listen to the opinions from other people while we make a choice. This scenario is not only true in the case of individual choice but today it is useful and right for organizations also. Very little computational study was carried out on opinions prior to the introduction of World Wide Web (WWW) due to limited availability of opinionated text for such analysis. In earlier days, when the person used to go for taking a decision, she usually used to ask for opinion from different sources either friends or relatives. As stated by Liu and Jawale when an organization is in need of opinions about their product or service from customer or general public, they generally conduct surveys or opinion polls from a group. Because of increase in the contents on the WWW through social media, the world has changed and became wealthy in data through advancement of Web. Currently people can put their reviews about products on respective business organizational sites and can express their opinions on nearly everything in various blogs and discussion forums. If anyone wants to
Purchase a product, there is no need to ask about the product to someone from friends and family. As many user reviews are easily available on the Web. So for the industries perhaps it is not required to conduct surveys to get customer opinions about their product or service or about their competitors because ample of information about the same is publicly available. Due to growth of internet and data contents on Web, searching the sites where opinions are available and then monitoring such sites on the internet is quite an intensive job because the opinion contents are available on different sites, and in turn every site may also

have large amount of opinions. It is possible that, in most of the cases, opinions or judgments about particular thing are not directly expressed. So it becomes difficult for a user to identify

such opinion related sites, read and extract opinion related contents, analyze such contents, get useful summary out of it, and use this summary to form an opinion. Thus, to do all these tasks there is need of automated opinion discovery and summarization system in the field of business decision making process to enhance the business strategies and business profit in the competitive world. It is observed in Liu that, the opinion contents which are available online on internet as well as off line is containing mostly textual information used by the customer to provide relevant product feedback. The information available in textual format can be generally classified as either facts or opinions. An objective expression about entity, objects or events and their attributes is known as facts. On the other hand, opinion is a subjective expression that describes sentiments of an individual, assessment of performance or emotions about entities, things or events and their attributes. Opinion is a broad concept. Majority of the current research study from Liu based on opinion mining concentrated on textual information processing.

In addition, it pays attention on opinion mining and retrieval of factual information provided and expressed in opinionated text. It also supports information retrieval, text classification, text mining and natural language processing. Processing of opinions was very little focused and studied concept to get exact opinions of the customer than factual information in recent research of sentiment analysis. Following section introduces the problem of sentiment analysis and its challenges in bringing automation in opinion mining system.

1.3 Previous Work

Sentiment analysis of natural language texts is a large and growing field. Previous work particularly relevant to our task falls naturally in two groups. The first relates to techniques to automatically generate sentiment lexicons. The second relates to systems that analyze sentiment (on a global or local basis) for entire documents.

Current research focuses on sentiment analysis of information gathered from social networking websites like Twitter, Facebook, MySpace to conclude viewers' response to a particular social event or issue. Sentiment analysis has endless applications like forecasting market movement based on news, blogs and social media. Currently, sentiment analysis is a very lucrative approach for hefty applications like 'Smart Cities'. These applications use methods based on document level and sentence level classification which use purely supervised or unsupervised classification algorithms. These algorithms are advanced by Fuzzy Formal Concept, Genetic Algorithms or Neural Networks by making them semi-supervised. Research also focused on sentiment analysis with networking to give a degree of parallelism. It focused on online accrued utility scheduling algorithm which gave them high speed on multiple processors. But this made the system much more complex. Research was also focused on Twitter sentiment analysis for security-related information gathering using normalized lexicon based sentiment analysis. While it provided a positive outcome, a universal dataset was not used. Current online product recommendation applications are comparing parameters like price, ratings and special offers on the product on different e-commerce websites and are not focusing on customers' personal experience by analyzing their reviews. Hence there is a need to develop a comprehensive application based on sentiment analysis which will give more importance to customer reviews.

Initial attempts to represent text used one-hot vectors to represent a word, where each dimension of the vector corresponds to a distinct word. And in the simple bag of words" representation of a document, the document is represented as the sum of the one-hot vectors of the words. However, these representations lose much of the semantic meaning that we associate with text. More recent work has focused on creating vector representations for both words and documents, with the idea of retaining as much information as possible. In word2vec, vector representations are computed for each word, with the result being that words whose meanings are related are generally closer in terms of Euclidean distance than between unrelated words. Deep learning models have also been effective in tackling this problem. Recurrent neural networks, by being able to retain memory between training examples, allows it to capture relations between words, within models such as LSTMs having the ability to remember important information across long stretches of time. In this project, we attempt to replicate some of these results.

1.4 IMDB Dataset

The labeled data set consists of 50,000 IMDB movie reviews, specially selected for sentiment analysis. The sentiment of reviews is binary, meaning the IMDB rating < 5 results in a sentiment score of 0, and rating >=7 have a sentiment score of 1. No individual movie has more than 30 reviews. The 25,000 review labeled training set does not include any of the same movies as the 25,000 review test set. In addition, there are another 50,000 IMDB reviews provided without any rating labels.

1.3.1 File Description:

- **labeledTrainData** - The labeled training set. The file is tab-delimited and has a header row followed by 25,000 rows containing an id, sentiment, and text for each review.

- **testData** - The test set. The tab-delimited file has a header row followed by 25,000 rows containing an id and text for each review. Your task is to predict the sentiment for each one.

- **unlabeledTrainData** - An extra training set with no labels. The tab-delimited file has a header row followed by 50,000 rows containing an id and text for each review.

- **sampleSubmission** - A comma-delimited sample submission file in the correct format.

1.3.2 Data Fields:

- **id** - Unique ID of each review
- **sentiment** - Sentiment of the review; 1 for positive reviews and 0 for negative reviews
- **review** - Text of the review

1.5 Environment Setup

Spyder (formerly Pydee) is an open source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates NumPy, SciPy, Matplotlib and IPython, as well as other open source software. It is released under the MIT license. Spyder is extensible with plugins, includes support for interactive tools for data inspection and embeds Python-specific code quality assurance and introspection instruments, such as Pyflakes, Pylint and Rope. It is available cross-platform through Anaconda, on Windows with WinPython and Python(x,y), on macOS through MacPorts, and on major Linux distributions such as Arch Linux, Debian, Fedora, Gentoo Linux, openSUSE and Ubuntu.

Spyder makes use of Qt either through the binding PyQt or PySide. This flexibility is reached through a small abstraction layer called QtPy.

Spyder is the Scientific Python Development Environment, a powerful interactive development environment for the Python language with advanced editing, interactive testing, debugging and introspection features and a numerical computing environment thanks to the support of *IPython* (enhanced interactive Python interpreter) and popular Python libraries such as *NumPy* (linear algebra), *SciPy* (signal and image processing) or *matplotlib* (interactive 2D/3D plotting). Spyder may also be used as a library providing powerful console-related widgets for your PyQt-based applications – for example, it may be used to integrate a debugging console directly in the layout of your graphical user interface.

Spyder is a Python development environment with a lot of features:

- **Editor**

  Multi-language editor with function/class browser, code analysis features (pyflakes and pylint are currently supported), code completion, horizontal and vertical splitting, and goto definition.

- **Interactive console**

  Python or IPython consoles with workspace and debugging support to instantly evaluate the code written in the Editor. It also comes with Matplotlib figures integration.

- **Documentation viewer**

  Show documentation for any class or function call made either in the Editor or a Console.

- **Variable explorer**

Explore variables created during the execution of a file. Editing them is also possible with several GUI based editors, like a dictionary and Numpy array ones.

- **Find in files**

  Supporting regular expressions and mercurial repositories

- **File explorer**
- **History log**

  Spyder may also be used as a PyQt5/PyQt4 extension library (module spyder). For example, the Python interactive shell widget used in Spyder may be embedded in your own PyQt5/PyQt4 application.

Chapter 2

REQUIREMENTS

2.1 Hardware Requirements

- 2 GB RAM
- Minimum 3 GB disk space
- 32 bit or 64 bit

2.2 Software Requirements

- Spyder IDE
- Python

Chapter 3

FLOW CHART

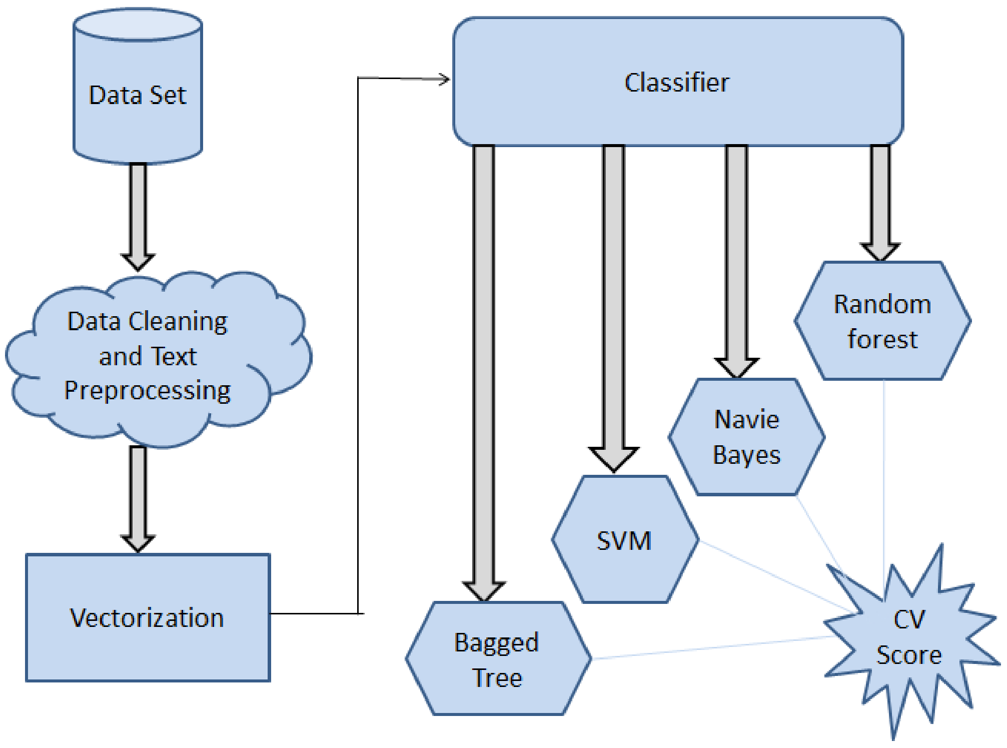


Fig 1.1 Sentiment Analysis Workflow

3.1 <u>Data Set</u>

Machine learning typically works with three data sets: training, dev and test. All three should randomly sample a larger body of data.

The first set you use is the **training set**, the largest of the three. Running a training set through a neural network teaches the net how to weigh different features, assigning them coefficients according to their likelihood of minimizing errors in your results.

The second set is your **test set**. It functions as a seal of approval, and you don't use it until the end. After you've trained and optimized your data, you test your neural net against this final random sampling. The results it produces should validate that your net accurately recognizes images, or recognizes them at least [x] percentage of them

3.2 <u>Training the Data</u>

Machine learning algorithms learn from data. It is critical that you feed them the right data for the problem you want to solve. Even if you have good data, you need to make sure that it is in a useful scale, format and even those meaningful features are included.

There are three step frameworks for data preparation and tactics:

Step 1: Data Selection

Consider what data is available, what data is missing and what data can be removed.

Step 2: Data Preprocessing

Organize your selected data by formatting, cleaning and sampling from it.

Step 3: Data Transformation

Transform preprocessed data ready for machine learning by engineering features using scaling, attribute decomposition and attribute aggregation.

Data preparation is a large subject that can involve a lot of iterations, exploration and analysis. Getting good at data preparation will make you a master at machine learning. For now, just consider the questions raised in this post when preparing data and always be looking for clearer ways of representing the problem you are trying to solve.

3.3 Classification

In machine learning and statistics, **classification** is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. An example would be assigning a given email into "spam" or "non-spam" classes or assigning a diagnosis to a given patient as described by observed characteristics of the patient (gender, blood pressure, presence or absence of certain symptoms, etc.). Classification is an example of pattern recognition.

In the terminology of machine learning, classification is considered an instance of supervised learning i.e. learning where a training set of correctly identified observations is available. The corresponding unsupervised procedure is known as clustering, and involves grouping data into categories based on some measure of inherent similarity or distance.

Often, the individual observations are analyzed into a set of quantifiable properties, known variously as explanatory variables or *features*. These properties may variously be categorical (e.g. "A", "B", "AB" or "O", for blood type, ordinal (e.g. "large", "medium" or "small"), integer-valued (e.g. the number of occurrences of a particular word in an email) or real-valued (e.g. a measurement of blood pressure). Other classifiers work by comparing observations to previous observations by means of a similarity or distance function.

An algorithm that implements classification, especially in a concrete implementation, is known as a **classifier**. The term "classifier" sometimes also refers to the mathematical function, implemented by a classification algorithm that maps input data to a category.

Examples of classification algorithms include:

- Linear classifiers
  - Fisher's linear discriminant
  - Logistic regression
  - Naive Bayes classifier
- Kernel estimation
  - k-nearest neighbor
- Boosting (meta-algorithm)
- Decision trees
- Random forests
- Neural networks
- Learning vector quantization

3.4 Test the Data

Once you have defined your problem and prepared your data you need to apply machine learning algorithms to the data in order to solve your problem. You can spend a lot of time choosing, running and tuning algorithms. You want to make sure you are using your time effectively to get closer to your goal.

3.5 Result

We will be evaluating the CV score for four classifiers (Naive-Bayes Classifier, Random Forest, Bagged Tree and SVM).

3.5.1 Cross Validation (CV) Score

Cross-validation, sometimes called **rotation estimation** is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. In a prediction problem, a model is usually given a dataset of *known data* on which training is run (*training dataset*), and a dataset of *unknown data* (or *first seen* data) against which the model is tested (*testing dataset*).[4] The goal of cross validation is to define a dataset to "test" the model in the training phase (i.e., the *validation dataset*) in order to limit problems like over fitting; give an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, for instance from a real problem), etc.

One round of cross-validation involves partitioning a sample of data into complementary subsets, performing the analysis on one subset (called the *training set*), and validating the analysis on the other subset (called the *validation set* or *testing set*). To reduce variability, multiple rounds of cross-validation are performed using different partitions, and the validation results are averaged over the rounds.

One of the main reasons for using cross-validation instead of using the conventional validation (e.g. partitioning the data set into two sets of 70% for training and 30% for test) is that there is not enough data available to partition it into separate training and test sets without losing significant modeling or testing capability. In these cases, a fair way to properly estimate model prediction performance is to use cross-validation as a powerful general technique.

Chapter 4

4.1 Data Cleaning and Preprocessing

Now, we have Read the data using pandas Library, Pre Processed it

4.1.1 Reading the data

We have a downloaded tsv file (unlabeledTrainData.tsv), which contains IMDB reviews, each with positive or negative sentiment label.

Next, read the tab-delimited file into python. To do this, we have used pandas package, which provides read_csv function for easily reading and writing data files.

4.1.2 Data Cleaning and Text Preprocessing

- Removing HTML Markup: First, we have removed the HTML tags. For this purpose, we have used the **Beautiful Soup** library

  Beautiful Soup Library: Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

- To remove punctuation and numbers, we will use a package for dealing with regular expressions, called **re**. The package comes built-in with Python; no need to install anything.

- We'll also convert our reviews to lower case and split them into individual words (called "tokenization” in NLP lingo)

- Removed Stop Words: Words such as "a", "and", "is", and "the” comes into category of stop words. Conveniently, there are Python packages that come with stop word lists built in. Now import a stop word list from the Python **Natural Language Toolkit** (NLTK) and Finally returned clean string or list.

4.2 Vectorization

Vectorization of a matrix is a linear transformation which converts the matrix into a column vector.

4.2.1 Words2vec

Word2vec is a group of related models that are used to produce word embedding. These models are shallow, two-layer neural networks that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.

Word2vec was created by a team of researchers led by Tomas Mikolov at Google. The algorithm has been subsequently analyzed and explained by other researchers. Embedding vectors created using the Word2vec algorithm have many advantages compared to earlier algorithms like Latent Semantic Analysis.

It was published by Google in 2013, is a neural network implementation that learns distributed representations for words. Other deep or recurrent neural network architectures had been proposed for learning word representations prior to this, but the major problem with these was the long time required to train the models. Word2vec learns quickly relative to other models.

Word2Vec does not need labels in order to create meaningful representations. This is useful, since most data in the real world is unlabeled. If the network is given enough training data (tens of billions of words), it produces word vectors with intriguing characteristics. Words with similar meanings appear in clusters, and clusters are spaced such that some word relationships, such as analogies, can be reproduced using vector math. The famous example is that, with highly trained word vectors, "king - man + woman = queen."

4.2.2 Term Frequency Inverse Document Frequency (TF-IDF)

In information retrieval, **tf–idf**, short for **term frequency–inverse document frequency**, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.It is often used as a weighting factor in information retrieval, text mining and user modeling. The tf-idf value increases proportionally to the number of times a word appears in the document, but is often offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. Nowadays, tf-idf is one of the most popular term-weighting schemes. For instance, 83% of text-based recommender systems in the domain of digital libraries use tf-idf.

Variations of the tf–idf weighting scheme are often used by search engines as a central tool in scoring and ranking a document's relevance given a user query. tf–idf can be successfully used for stop-words filtering in various subject fields including text summarization and classification.One of the simplest ranking functions is computed by summing the tf–idf for each query term; many more sophisticated ranking functions are variants of this simple model.

Typically, the tf-idf weight is composed by two terms: the first computes the normalized Term Frequency (TF), the number of times a word appears in a document, divided by the total number of words in that document; the second term is the Inverse Document Frequency (IDF), computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears.

- **TF: Term Frequency**, which measures how frequently a term occurs in a document. Since every document is different in length, it is possible that a term would appear much more times in long documents than shorter ones. Thus, the term frequency is often divided by the document length (aka. the total number of terms in the document) as a way of normalization: TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document).
- **IDF: Inverse Document Frequency**, which measures how important a term is. While computing TF, all terms are considered equally important. However it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus we need to weigh down the frequent terms while scale up the rare ones, by computing the following: IDF(t) = loge(Total number of documents / Number of documents with term t in it).

4.3 Feature Scaling

Feature scaling is a method used to standardize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step.

Features scaling though standardization (or Z-score normalization) can be an importance preprocessing step for many machine learning algorithms. Standardization involves rescaling the features such that they'll have the properties of a standard normal distribution with a mean of zero and a standard deviation of one.

While many algorithms (such as SVM, K-nearest neighbors and logistic regression) require features to be normalized, intuitively we can think of Principle Component Analysis (PCA) as being a prime example of when normalization is important. In PCA we are interested in the components that maximize the variance. If there exists components (e.g human height) that vary less than other components (e.g human weight) because of their respective scales (meters vs. kilos) it can be seen how not scaling the features would cause PCA to determine that the direction of maximal variance more closely corresponds with the 'weight' axis. As a change in height of one meter can be considered much more important than the change in weight of one kilogram, it is easily seen that this determination is incorrect. In the case of PCA, scaling features using normalization is preferred over using min-max scaling as the primary components are computed using the correlation matrix as opposed to the covariance matrix.

Since the range of values of raw data varies widely, in some machine learning algorithms, objective functions will not work properly without normalization. For example, the majority of classifiers calculate the distance between two points by the distance. If one of the features has a broad range of values, the distance will be governed by this particular feature. Therefore, the range of all features should be normalized so that each feature contributes approximately proportionately to the final distance. Another reason why feature scaling is applied is that gradient descent converges much faster with feature scaling than without it.

Few advantages of normalizing the data are as follows:

- It makes your training faster.
- It prevents you from getting stuck in local optima.
- It gives you a better error surface shape.
- Weight decay and bayes optimization can be done more conveniently.

However, there are few algorithms such as Logistic Regression and Decision Trees that are not affected by scaling of input data.

4.4 Dimension Reduction

In machine learning and statistics, **dimensionality reduction** or **dimension reduction** is the process of reducing the number of random variables under consideration, via obtaining a set of principal variables.

High dimensional data is often found in different disciplines when doing data analysis, and each disciplines may have its specific demand to perform dimensionality reduction. In this section, general reasons and requirements of dimension reduction are presented, both in practical and theoretical perspectives. Indeed, dimension reduction can focus only on the observed data set and ignores the generalization performance, as used in data compression. The reasons for dimension reduction are:

- **Redundancy reduction and intrinsic structure discovery**: In multimedia researches, the processing data is naturally of high dimensionality, such as digital signals, digital audios, and digital videos. In these cases, each feature usually serves only as a simple measurement, while combining all of them can express a complicated concept or perception. And even if some features are missed, the concept could still be constructed from the remaining features, which means there is partial redundancy and dependency across features. In some applications, the existing redundancy is desired to be removed in order to save the usage of memory and the cost of transmission, such as image, audio, speech, video compression.

- **Intrinsic structure discovery**: Following the idea that there is some redundancy across features, in some other applications, these observed or measured features are assumed coming from a much lower dimensional intrinsic structure. And dimensionality reduction techniques are performed to discover the intrinsic structure or latent variables (the intrinsic parameters or degree of freedom) which can better explain a complicated phenomenon or concept, such as the use of independent component analysis (ICA) in multi-channel electroencephalographic (EEG) data, the use of probabilistic latent semantic analysis (PLSA) in text information retrieval.

- **Removal of irrelevant and noisy features**: During the feature extraction process, we desire to preserve as many information as possible from the original raw data. These extracted features seems meaningful from our perspective, while may not convey explicit correlations with the ongoing task. For example, the Haar feature is useful in the task of face detection, but not in the task of face recognition and age estimation. Besides, raw data and the feature extraction process may suffer from noise and degrades the performance of learning. Dimensionality reduction techniques have been used to solve these problems, for example, the Adaboost has intrinsic power to select the most relevant features for the ongoing task, and the PCA has been used to remove noise in face patches.

- **Feature extraction**: In many pattern recognition tasks, especially for the tasks of object recognition and classification, dimensionality reduction is used for feature extraction process. Compared to the reason of intrinsic structure discovery, the usage here doesn't explicitly assume the existing of an intrinsic parameter space, but want to simultaneously lower down the dimensionality of raw data and preserve the most compact representation of information from it. The use of DR techniques in face-related researches is mainly of this reason.

- **Visualization purpose**: One of the best ways for human to understand a concept and the relationship between feature samples is visualization in 2D or 3D representation, so how to preserve the structure in high dimensional feature space into 2D or 3D space is an interesting and challenging problem. The Isomap and LLE which will be introduced later show the ability to locate face patches in a 2D plane and preserve the appearance similarities.

- **Computation and Machine learning perspective**: For applications with real-time usage or with limited computational resources, dimensionality reduction is often required, although some information will get lost. And the last but probably the most important reason, the feature dimensionality has positive correlations with VC dimension and model complexity, so under limited training samples, we also need to limit the feature dimensionality to maintain the generalization performance.

Dimension Reduction can be divided into feature selection and feature extraction.

4.4.1 Feature Selection:

In machine learning and statistics, feature selection, also known as variable selection, attribute selection or variable subset selection, is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. Feature selection techniques are used for four reasons:

- simplification of models to make them easier to interpret by researchers/users
- shorter training times
- to avoid the curse of dimensionality
- enhanced generalization by reducing overfitting (formally, reduction of variance)

The central premise when using a feature selection technique is that the data contains many features that are either redundant or irrelevant, and can thus be removed without incurring much loss of information. Redundant or irrelevant features are two distinct notions, since one relevant feature may be redundant in the presence of another relevant feature with which it is strongly correlated.

Feature selection techniques should be distinguished from feature extraction. Feature extraction creates new features from functions of the original features, whereas feature selection returns a subset of the features. Feature selection techniques are often used in domains where there are many features and comparatively few samples (or data points). Archetypal cases for the application of feature selection include the analysis of written texts and DNA microarray data, where there are many thousands of features, and a few tens to hundreds of samples.

4.4.2 <u>Feature Extraction</u>:

In machine learning, pattern recognition and in image processing, feature extraction starts from an initial set of measured data and builds derived values (features) intended to be informative and non-redundant, facilitating the subsequent learning and generalization steps, and in some cases leading to better human interpretations. Feature extraction is related to dimensionality reduction.

When the input data to an algorithm is too large to be processed and it is suspected to be redundant (e.g. the same measurement in both feet and meters, or the repetitiveness of images presented as pixels), then it can be transformed into a reduced set of features (also named a feature vector). Determining a subset of the initial features is called feature selection. The selected features are expected to contain the relevant information from the input data, so that the desired task can be performed by using this reduced representation instead of the complete initial data. Feature extraction a type of dimensionality reduction that efficiently represents interesting parts of an image as a compact feature vector. This approach is useful when image sizes are large and a reduced feature representation is required to quickly complete tasks such as image matching and retrieval. Feature detection, feature extraction, and matching are often combined to solve common computer vision problems such as object detection and recognition, content-based image retrieval, face detection and recognition, and texture classification.

4.4.3 Chi-square dimension reduction:

The chi-square test is a statistical test of independence to determine the dependency of two variables. It shares similarities with coefficient of determination, R^2R^2. However, chi-square test is only applicable to categorical or nominal data while R^2R^2 is only applicable to numeric data.

Chi Square Test is used in statistics to test the independence of two events. Given dataset about two events, we can get the observed count O and the expected count E. Chi Square Score measures how much the expected counts E and observed Count O derivate from each other. In feature selection, the two events are occurrence of the feature and occurrence of the class.

In other words, we want to test whether the occurrence of a specific feature and the occurrence of a specific class are independent.

If the two events are dependent, we can use the occurrence of the feature to predict the occurrence of the class. We aim to select the features, of which the occurrence is highly dependent on the occurrence of the class. When the two events are independent, the observed count is close to the expected count, thus a small chi square score. So a high value of \chi^2 indicates that the hypothesis of independence is incorrect. In other words, the higher value of the \chi^2 core, the more likelihood the feature is correlated with the class, thus it should be selected for model training.

From the definition, of chi-square we can easily deduce the application of chi-square technique in feature selection. Suppose you have a target variable (i.e., the class label) and some other features (feature variables) that describes each sample of the data. Now, we calculate chi-square statistics between every feature variable and the target variable and observe the existence of a relationship between the variables and the target. If the target variable is independent of the feature variable, we can discard that feature variable. If they are dependent, the feature variable is very important.

4.4.4 Singular Value Decomposition (SVD):

The singular value decomposition of a matrix A is the factorization of A into the product of three matrices A=UDVT where the columns of U and V are orthonormal and the matrix D is diagonal with positive real entries. The SVD is useful in many tasks. Here we mention some examples. First, in many applications, the data matrix A is close to a matrix of low rank and it is useful to find a low rank matrix which is a good approximate onto the data matrix. We will show that from the singular value decomposition of A, we can get the matrix of rank k which best approximates A; in fact we can do this for every k. Also, singular value decomposition is defined for all matrices (rectangular or square) unlike the more commonly used spectral decomposition in Linear Algebra. The reader familiar with eigenvectors and eigenvalues (we do not assume familiarity here) will also realize that we need conditions on the matrix to ensure orthogonality of eigenvectors. In contrast, the columns of V in the singular value decomposition, called the right singular vectors of A, always form an orthogonal set with no assumptions on A. The columns of U are called the left singular vectors and they also form an orthogonal set. A simple consequence of the orthogonality is that for a square and invertible matrix A, the inverse of A is V D−1UT, as the reader can verify. To gain insight into the SVD, treat the rows of a nn×d matrix A as n points in ad-dimensional space and consider the problem of finding the best k-dimensional subspace with respect to the set of points. Here best means minimize the sum of the squares of the perpendicular distances of the points to the subspace. We begin with a special case of the problem where the subspace is 1-dimensional, a line through the origin. Finding the best fitting line through the origin with respect to a set of points{xi|1≤i≤n}in the plane means minimizing the sum of the squared distances of the points to the line. Here distance is measured perpendicular to the line. The problem is called the best least squares fit. In the best least squares fit, one is minimizing the distance to a subspace. An alternative problem is to find the function that best fits some data. Here one variable y is a function of the variables x1, x2,...,xd and one wishes to minimize the vertical distance ,i.e., distance in the y direction, to the subspace of the xi rather than minimize the perpendicular distance to the subspace being fit to the data.

4.5 Classifier

4.5.1 Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set.

The first algorithm for random decision forests was created by Tin Kam Ho using the random subspace method which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.

Features of Random Forests are:

- It is unexcelled in accuracy among current algorithms.
- It runs efficiently on large data bases.
- It can handle thousands of input variables without variable deletion.
- It gives estimates of what variables are important in the classification.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.
- It has methods for balancing error in class population unbalanced data sets.
- Generated forests can be saved for future use on other data.
- Prototypes are computed that give information about the relation between the variables and the classification.

- It computes proximities between pairs of cases that can be used in clustering, locating outliers, or (by scaling) give interesting views of the data.
- The capabilities of the above can be extended to unlabeled data, leading to unsupervised clustering, data views and outlier detection.
- It offers an experimental method for detecting variable interactions.

Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of overcoming over-fitting problem of individual decision tree.

In other words, random forests are an ensemble learning method for classification and regression that operate by constructing a lot of decision trees at training time and outputting the class that is the mode of the classes output by individual trees. Random Forest is one of the most widely used machine learning algorithm for classification. It can also be used for regression model (i.e. continuous target variable) but it mainly performs well on classification model (i.e. categorical target variable). It has become a lethal weapon of modern data scientists to refine the predictive model. The best part of the algorithm is that there are a very few assumptions attached to it so data preparation is less challenging and results to time saving. It's listed as a top algorithm (with ensembling) in Kaggle Competitions.

Random Forest algorithm works in the following way:

1. **Random Record Selection:** Each tree is trained on roughly 2/3rd of the total training data (exactly 63.2%). Cases are drawn at random with replacement from the original data. This sample will be the training set for growing the tree.

2. **Random Variable Selection:** Some predictor variables (say, m) are selected at random out of all the predictor variables and the best split on these m is used to split the node.

   By default, m is square root of the total number of all predictors for classification. For regression, m is the total number of all predictors divided by 3.The value of m is held constant during the forest growing. In a standard tree, each split is created after examining every variable and picking the best split from all the variables.

3. For each tree, using the leftover (36.8%) data, calculate the misclassification rate - out of bag (OOB) error rate. Aggregate error from all trees to determine overall OOB error rate for the classification.

4. Each tree gives a classification, and we say the tree "votes" for that class. The forest chooses the classification having the most votes over all the trees in the forest. For a binary dependent variable, the vote will be YES or NO, count up the YES votes. This is the RF score and the percent YES votes received is the predicted probability. In regression case, it is average of dependent variable.

The shortcomings of Random Forest are:

- Random Forests aren't good at generalizing to cases with completely new data. For example, if I tell you that 1 chocolate costs $1, 2 chocolates cost $2, and 3 chocolates cost $3, how much do 10 chocolates cost? A linear regression can easily figure this out, while a Random Forest has no way of finding the answer.

- If a variable is a categorical variable with multiple levels, random forests are biased towards the variable having multiple levels.

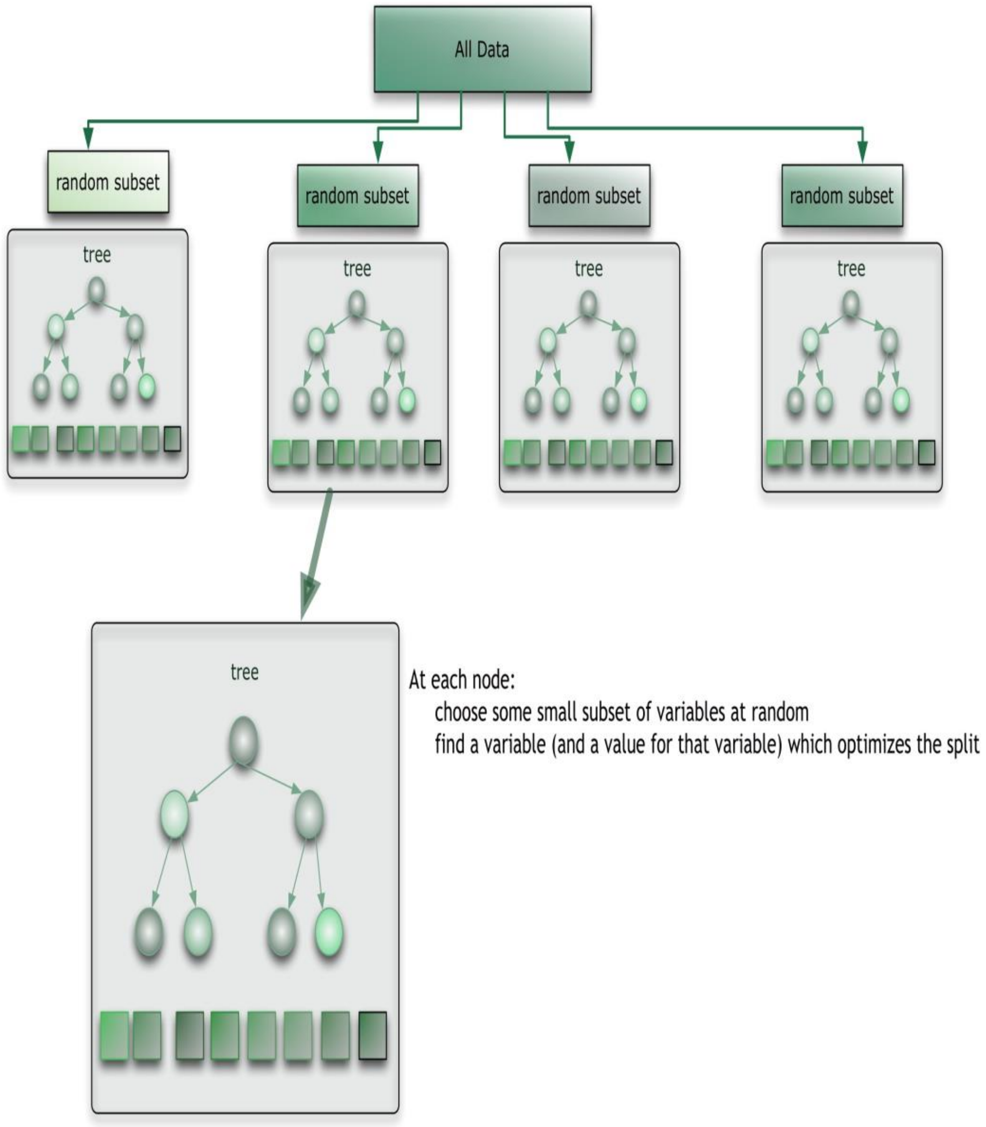

All Data
random subset
random subset
random subset
random subset
tree
tree
tree
tree
tree
At each node:
choose some small subset of variables at random
find a variable (and a value for that variable) which optimizes the split

Fig 1.2 Random Forest Algorithm

4.5.2 Support Vector Machines (SVM)

In machine learning **support vector machines** (**SVMs**, also **support vector networks**) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

When data are not labeled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups. The clustering algorithm which provides an improvement to the support vector machines is called **support vector clustering** and is often used in industrial applications either when data are not labeled or when only some data are labeled as a preprocessing for a classification pass. More formally, a support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.

The advantages of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

- If the number of features is much greater than the number of samples, the method is likely to give poor performances.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

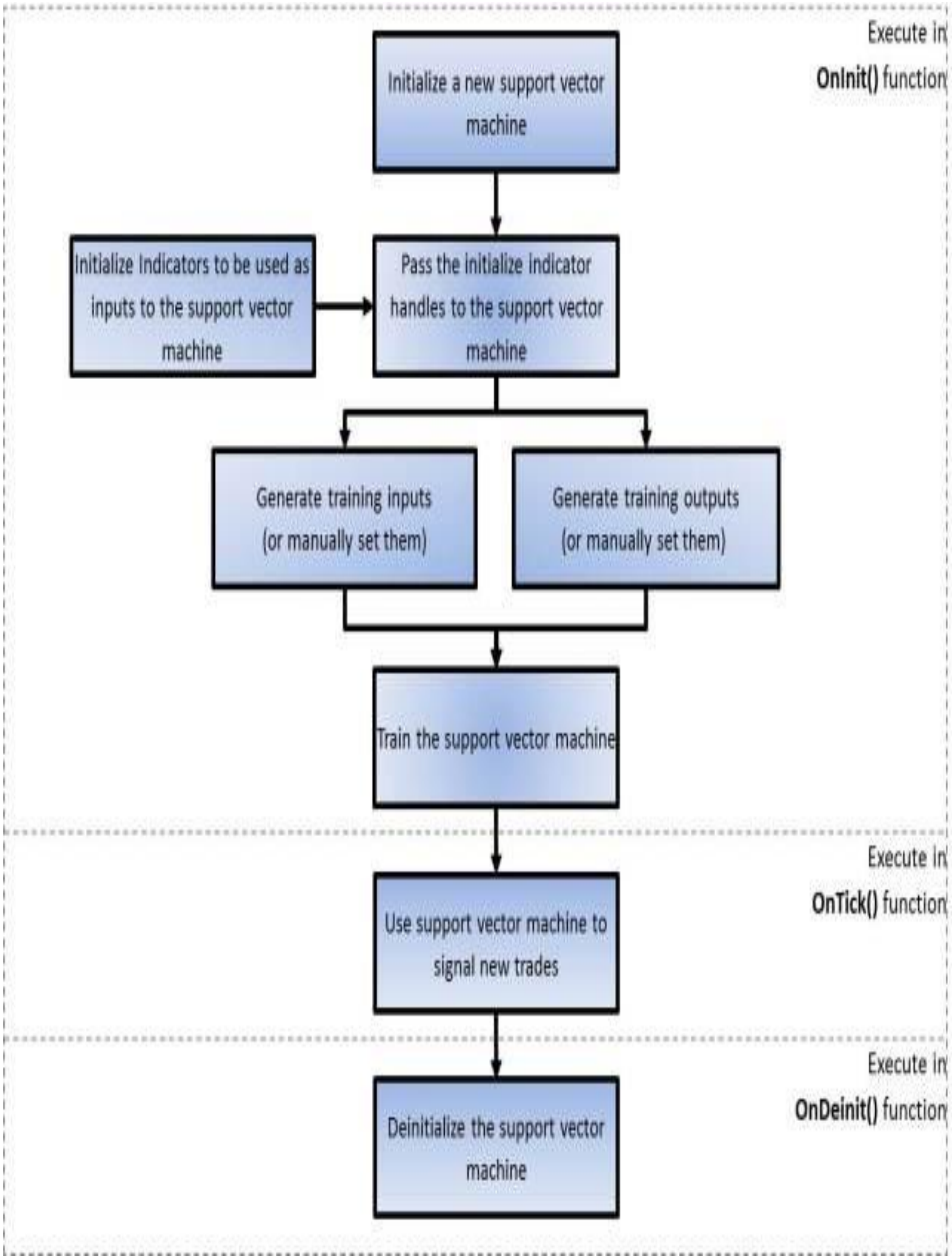

Execute in
OnInit() function
Initialize a new support vector machine
Initialize Indicators to be used as inputs to the support vector machine
Pass the initialize indicator handles to the support vector machine
Generate training inputs (or manually set them)
Generate training outputs (or manually set them)
Train the support vector machine
Execute in
OnTick() function
Use support vector machine to signal new trades
Execute in
OnDeinit() function
Deinitialize the support vector machine

Fig 1.3 Support Vector Machine

4.5.3 Naive Bayes Classifier

In machine learning, **naive Bayes classifiers** are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong independence assumptions between the features. Naive Bayes has been studied extensively since the 1950s. It was introduced under a different name into the text retrieval community in the early 1960s, and remains a popular (baseline) method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, sports or politics, etc.) with word frequencies as the features. With appropriate pre-processing, it is competitive in this domain with more advanced methods including support vector machines. It also finds application in automatic medical diagnosis.

Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers. In the statistics and computer science literature, Naive Bayes models are known under a variety of names, including **simple Bayes** and **independence Bayes**. All these names reference the use of Bayes' theorem in the classifier's decision rule, but naive Bayes is not (necessarily) a Bayesian method.

Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values. It is called *naive Bayes* because the calculation of the probabilities for each hypothesis is simplified to make their calculation tractable. Rather than attempting to calculate the values of each attribute value P(d1, d2, d3|h), they are assumed to be conditionally independent given the target value and calculated as $P(d1|h)$ * $P(d2|H)$ and so on. This is a very strong assumption that is most unlikely in real data, i.e. that the attributes do not interact. Nevertheless, the approach performs surprisingly well on data where this assumption do not hold.

The Naïve Bayesian classifier works as follows: Suppose that there exist a set of training data, D, in which each tuple is represented by an n-dimensional feature vector, $X=x_1,x_2,..,x_n$, indicating n measurements made on the tuple from n attributes or features. Assume that there are m classes, $C_1,C_2,...,C_m$. Given a tuple X, the classifier will predict that X belongs to C_i if and only if: $P(C_i|X)>P(C_j|X)$, where $i,j\in[1,m]$a n d $i\neq j$. $P(C_i|X)$ is computed as:

$$\mathbf{P(C_i|X)=\prod_{k=1}^{n}P(x_k|C_i)}$$

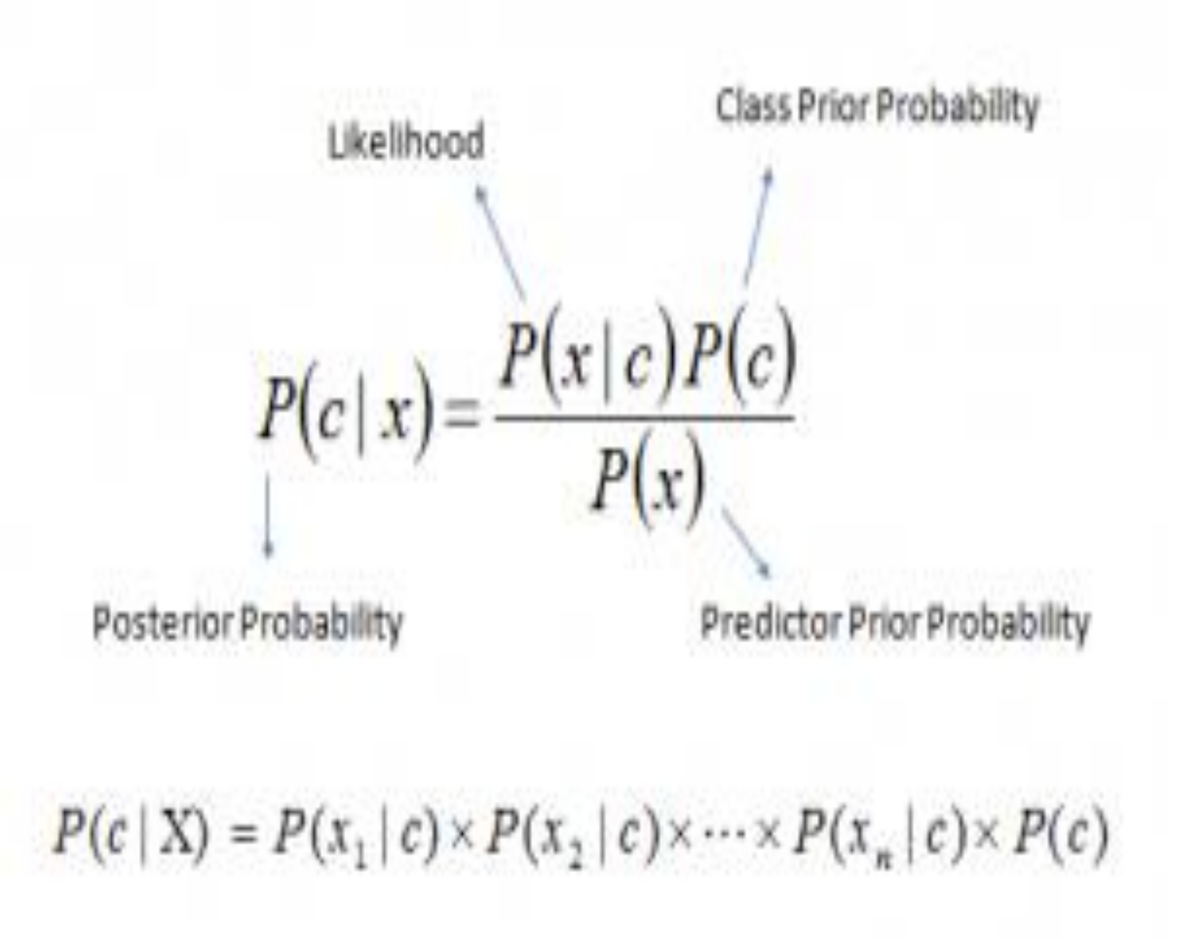
Advantages of Naïve-Bayes classifier are:

- 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).

Disadvantages of Naïve-Bayes classifier are:

- 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 is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.



$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

<u>Fig 1.4 Naïve Bayes Theorem</u>

4.5.4 <u>Bagged Tree</u>

Bootstrap aggregating, also called bagging, is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid overfitting. Although it is usually applied to decision tree methods, it can be used with any type of method. Bagging is a special case of the model averaging approach.

Bootstrap Aggregation (or Bagging for short), is a simple and very powerful ensemble method. An ensemble method is a technique that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model.

Bootstrap Aggregation is a general procedure that can be used to reduce the variance for those algorithms that have high variance. An algorithm that has high variance are decision trees, like classification and regression trees (CART).

Decision trees are sensitive to the specific data on which they are trained. If the training data is changed (e.g. a tree is trained on a subset of the training data) the resulting decision tree can be quite different and in turn the predictions can be quite different.

Bagging is the application of the Bootstrap procedure to a high-variance machine learning algorithm, typically decision trees.

Let’s assume we have a sample dataset of 1000 instances (x) and we are using the CART algorithm. Bagging of the CART algorithm would work as follows.

Create many (e.g. 100) random sub-samples of our dataset with replacement.

Train a CART model on each sample.

Given a new dataset, calculate the average prediction from each model.

For example, if we had 5 bagged decision trees that made the following class predictions for a in input sample: blue, blue, red, blue and red, we would take the most frequent class and predict blue.

When bagging with decision trees, we are less concerned about individual trees overfitting the training data. For this reason and for efficiency, the individual decision trees are grown deep (e.g. few training samples at each leaf-node of the tree) and the trees are not pruned. These trees will have both high variance and low bias. These are important characterize of sub-models when combining predictions using bagging.

The only parameters when bagging decision trees is the number of samples and hence the number of trees to include. This can be chosen by increasing the number of trees on run after run until the accuracy begins to stop showing improvement (e.g. on a cross validation test harness). Very large numbers of models may take a long time to prepare, but will not overfit the training data.

Just like the decision trees themselves, Bagging can be used for classification and regression problems.

Chapter-5

PERFORMANCE EVALUATION

Using Chi-square dimension reduction for Random Forest, Naïve-Bayes and Bagged Tree whereas using SVD (Singular Value Decomposition) for SVM (Support Vector Machine).

	TFIDF	Binary	Int
Random Forest	0.8304	0.82704	0.8256
SVM	0.88624	0.87	0.86488
Naive Bayes	0.87236	0.86432	0.83704
Bagged Tree	0.81112	0.80976	0.80488

Table for different CV Scores using different classifiers

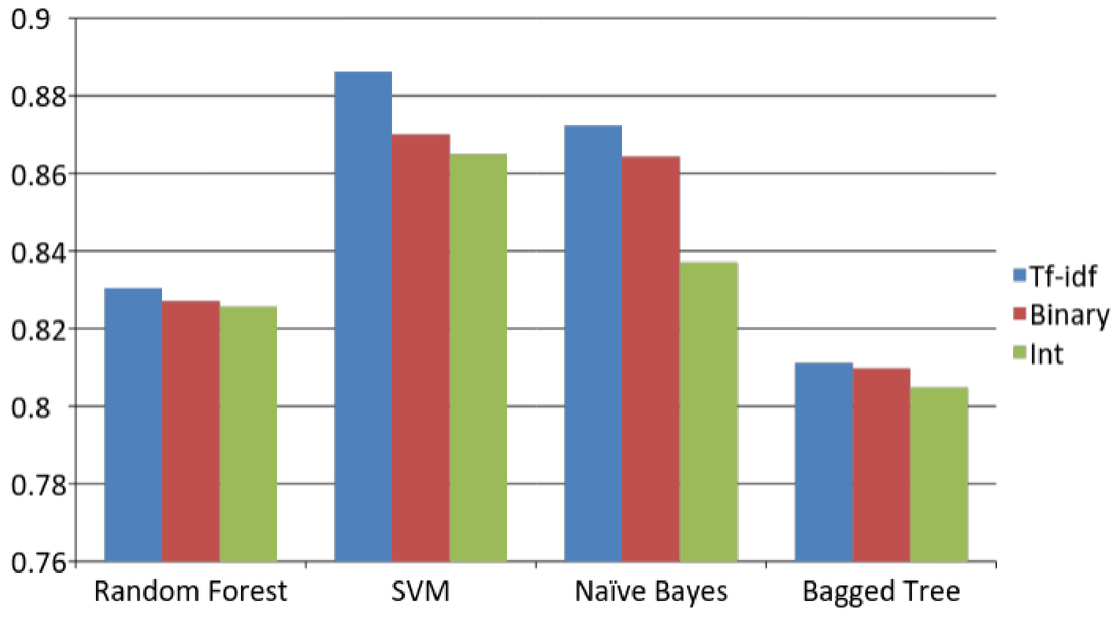
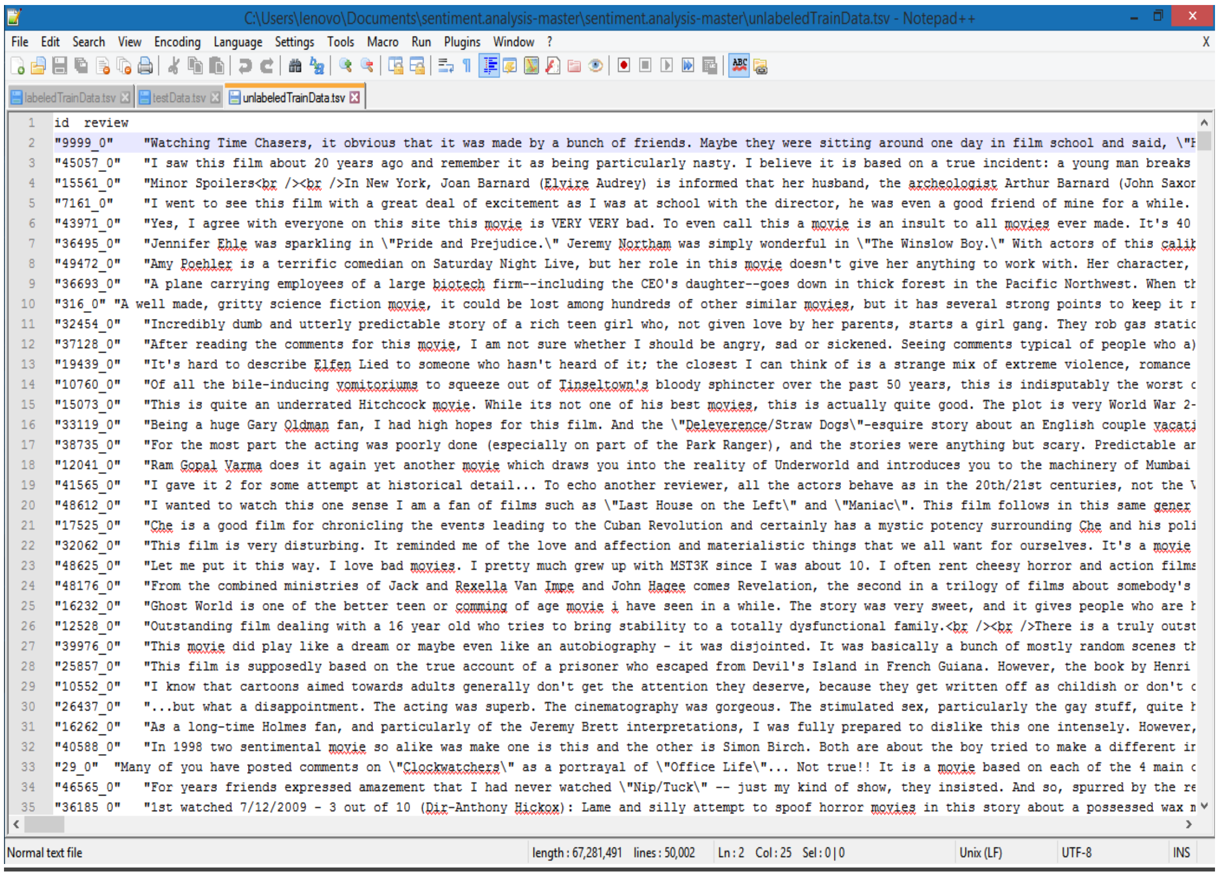


Fig 1.5 CV Score Bar Graph

Chapter-6

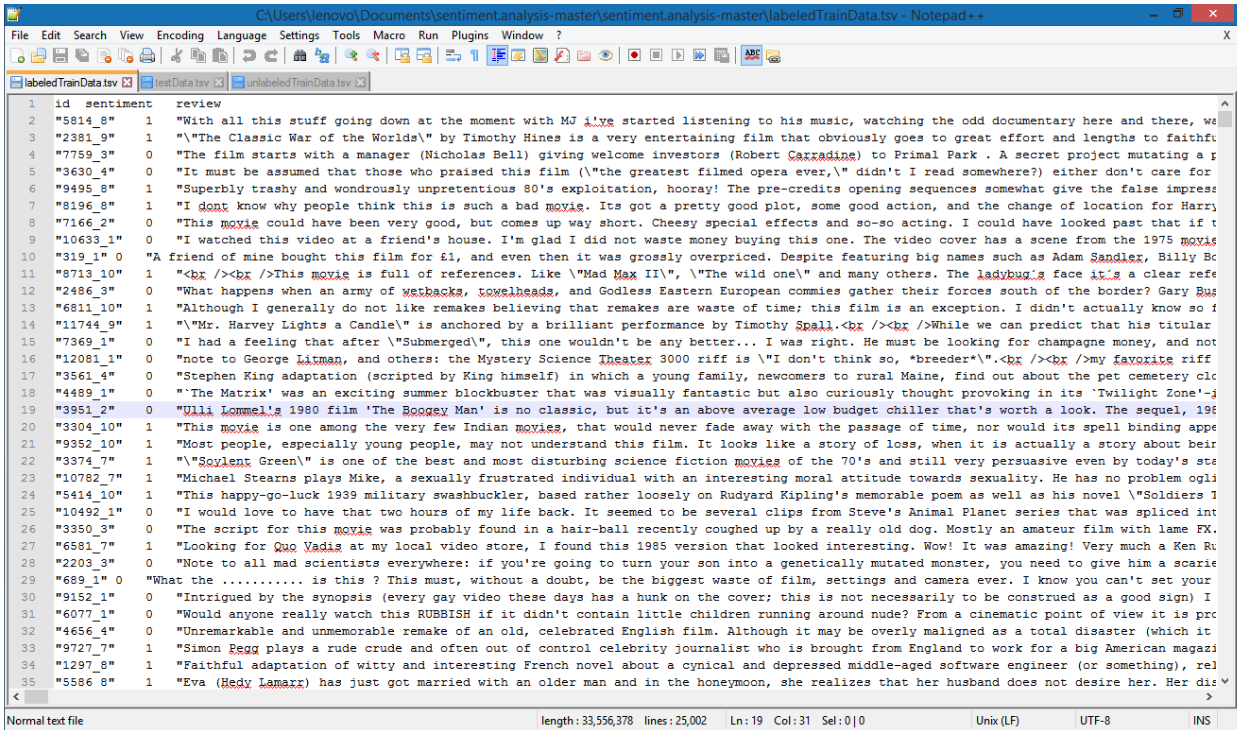
Screenshots



```
id  review
"9999_0"    "Watching Time Chasers, it obvious that it was made by a bunch of friends. Maybe they were sitting around one day in film school and said, \"F
"45057_0"   "I saw this film about 20 years ago and remember it as being particularly nasty. I believe it is based on a true incident: a young man breaks
"15561_0"   "Minor Spoilers<br /><br />In New York, Joan Barnard (Elvire Audrey) is informed that her husband, the archeologist Arthur Barnard (John Saxon
"7161_0"    "I went to see this film with a great deal of excitement as I was at school with the director, he was even a good friend of mine for a while.
"43971_0"   "Yes, I agree with everyone on this site this movie is VERY VERY bad. To even call this a movie is an insult to all movies ever made. It's 40
"36495_0"   "Jennifer Ehle was sparkling in \"Pride and Prejudice.\" Jeremy Northam was simply wonderful in \"The Winslow Boy.\" With actors of this calib
"49472_0"   "Amy Poehler is a terrific comedian on Saturday Night Live, but her role in this movie doesn't give her anything to work with. Her character,
"36693_0"   "A plane carrying employees of a large biotech firm--including the CEO's daughter--goes down in thick forest in the Pacific Northwest. When th
"316_0" "A well made, gritty science fiction movie, it could be lost among hundreds of other similar movies, but it has several strong points to keep it r
"32454_0"   "Incredibly dumb and utterly predictable story of a rich teen girl who, not given love by her parents, starts a girl gang. They rob gas static
"37128_0"   "After reading the comments for this movie, I am not sure whether I should be angry, sad or sickened. Seeing comments typical of people who a)
"19439_0"   "It's hard to describe Elfen Lied to someone who hasn't heard of it; the closest I can think of is a strange mix of extreme violence, romance
"10760_0"   "Of all the bile-inducing vomitoriums to squeeze out of Tinseltown's bloody sphincter over the past 50 years, this is indisputably the worst c
"15073_0"   "This is quite an underrated Hitchcock movie. While its not one of his best movies, this is actually quite good. The plot is very World War 2-
"33119_0"   "Being a huge Gary Oldman fan, I had high hopes for this film. And the \"Deleverence/Straw Dogs\"-esquire story about an English couple vacati
"38735_0"   "For the most part the acting was poorly done (especially on part of the Park Ranger), and the stories were anything but scary. Predictable ar
"12041_0"   "Ram Gopal Varma does it again yet another movie which draws you into the reality of Underworld and introduces you to the machinery of Mumbai
"41565_0"   "I gave it 2 for some attempt at historical detail... To echo another reviewer, all the actors behave as in the 20th/21st centuries, not the \
"48612_0"   "I wanted to watch this one sense I am a fan of films such as \"Last House on the Left\" and \"Maniac\". This film follows in this same gener
"17525_0"   "Che is a good film for chronicling the events leading to the Cuban Revolution and certainly has a mystic potency surrounding Che and his poli
"32062_0"   "This film is very disturbing. It reminded me of the love and affection and materialistic things that we all want for ourselves. It's a movie
"48625_0"   "Let me put it this way. I love bad movies. I pretty much grew up with MST3K since I was about 10. I often rent cheesy horror and action films
"48176_0"   "From the combined ministries of Jack and Rexella Van Impe and John Hagee comes Revelation, the second in a trilogy of films about somebody's
"16232_0"   "Ghost World is one of the better teen or comming of age movie i have seen in a while. The story was very sweet, and it gives people who are h
"12528_0"   "Outstanding film dealing with a 16 year old who tries to bring stability to a totally dysfunctional family.<br /><br />There is a truly outst
"39976_0"   "This movie did play like a dream or maybe even like an autobiography - it was disjointed. It was basically a bunch of mostly random scenes th
"25857_0"   "This film is supposedly based on the true account of a prisoner who escaped from Devil's Island in French Guiana. However, the book by Henri
"10552_0"   "I know that cartoons aimed towards adults generally don't get the attention they deserve, because they get written off as childish or don't c
"26437_0"   "...but what a disappointment. The acting was superb. The cinematography was gorgeous. The stimulated sex, particularly the gay stuff, quite h
"16262_0"   "As a long-time Holmes fan, and particularly of the Jeremy Brett interpretations, I was fully prepared to dislike this one intensely. However,
"40588_0"   "In 1998 two sentimental movie so alike was make one is this and the other is Simon Birch. Both are about the boy tried to make a different ir
"29_0"  "Many of you have posted comments on \"Clockwatchers\" as a portrayal of \"Office Life\"... Not true!! It is a movie based on each of the 4 main c
"46565_0"   "For years friends expressed amazement that I had never watched \"Nip/Tuck\" -- just my kind of show, they insisted. And so, spurred by the re
"36185_0"   "1st watched 7/12/2009 - 3 out of 10 (Dir-Anthony Hickox): Lame and silly attempt to spoof horror movies in this story about a possessed wax m
```

C:\Users\lenovo\Documents\sentiment.analysis-master\sentiment.analysis-master\unlabeledTrainData.tsv - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
labeledTrainData.tsv
testData.tsv
unlabeledTrainData.tsv
Normal text file
length : 67,281,491 lines : 50,002
Ln : 2 Col : 25 Sel : 0 | 0
Unix (LF)
UTF-8
INS

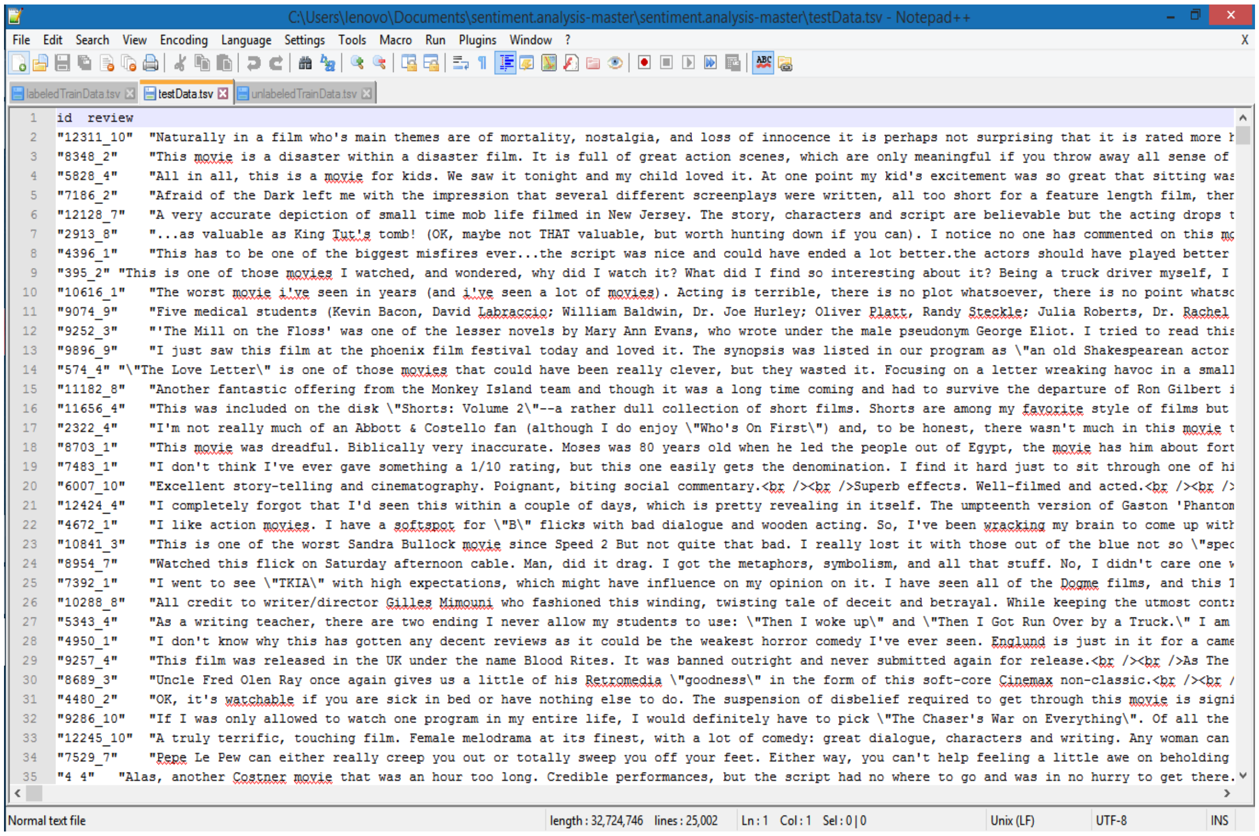
Fig 1.6 Unlabeled Train data


C:\Users\lenovo\Documents\sentiment.analysis-master\sentiment.analysis-master\labeledTrainData.tsv - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
labeledTrainData.tsv
testData.tsv
unlabeledTrainData.tsv

```
id  sentiment   review
"5814_8"    1   "With all this stuff going down at the moment with MJ i've started listening to his music, watching the odd documentary here and there, wa
"2381_9"    1   "\"The Classic War of the Worlds\" by Timothy Hines is a very entertaining film that obviously goes to great effort and lengths to faithfu
"7759_3"    0   "The film starts with a manager (Nicholas Bell) giving welcome investors (Robert Carradine) to Primal Park . A secret project mutating a p
"3630_4"    0   "It must be assumed that those who praised this film (\"the greatest filmed opera ever,\" didn't I read somewhere?) either don't care for
"9495_8"    1   "Superbly trashy and wondrously unpretentious 80's exploitation, hooray! The pre-credits opening sequences somewhat give the false impress
"8196_8"    1   "I dont know why people think this is such a bad movie. Its got a pretty good plot, some good action, and the change of location for Harry
"7166_2"    0   "This movie could have been very good, but comes up way short. Cheesy special effects and so-so acting. I could have looked past that if t
"10633_1"   0   "I watched this video at a friend's house. I'm glad I did not waste money buying this one. The video cover has a scene from the 1975 movie
"319_1" 0   "A friend of mine bought this film for £1, and even then it was grossly overpriced. Despite featuring big names such as Adam Sandler, Billy Bc
"8713_10"   1   "<br /><br />This movie is full of references. Like \"Mad Max II\", \"The wild one\" and many others. The ladybug´s face it´s a clear refe
"2486_3"    0   "What happens when an army of wetbacks, towelheads, and Godless Eastern European commies gather their forces south of the border? Gary Bus
"6811_10"   1   "Although I generally do not like remakes believing that remakes are waste of time; this film is an exception. I didn't actually know so f
"11744_9"   1   "\"Mr. Harvey Lights a Candle\" is anchored by a brilliant performance by Timothy Spall.<br /><br />While we can predict that his titular
"7369_1"    0   "I had a feeling that after \"Submerged\", this one wouldn't be any better... I was right. He must be looking for champagne money, and not
"12081_1"   0   "note to George Litman, and others: the Mystery Science Theater 3000 riff is \"I don't think so, *breeder*\".<br /><br />my favorite riff
"3561_4"    0   "Stephen King adaptation (scripted by King himself) in which a young family, newcomers to rural Maine, find out about the pet cemetery clc
"4489_1"    0   "`The Matrix' was an exciting summer blockbuster that was visually fantastic but also curiously thought provoking in its `Twilight Zone'-j
"3951_2"    0   "Ulli Lommel's 1980 film 'The Boogey Man' is no classic, but it's an above average low budget chiller that's worth a look. The sequel, 198
"3304_10"   1   "This movie is one among the very few Indian movies, that would never fade away with the passage of time, nor would its spell binding appe
"9352_10"   1   "Most people, especially young people, may not understand this film. It looks like a story of loss, when it is actually a story about beir
"3374_7"    1   "\"Soylent Green\" is one of the best and most disturbing science fiction movies of the 70's and still very persuasive even by today's sta
"10782_7"   1   "Michael Stearns plays Mike, a sexually frustrated individual with an interesting moral attitude towards sexuality. He has no problem ogli
"5414_10"   1   "This happy-go-luck 1939 military swashbuckler, based rather loosely on Rudyard Kipling's memorable poem as well as his novel \"Soldiers T
"10492_1"   0   "I would love to have that two hours of my life back. It seemed to be several clips from Steve's Animal Planet series that was spliced int
"3350_3"    0   "The script for this movie was probably found in a hair-ball recently coughed up by a really old dog. Mostly an amateur film with lame FX.
"6581_7"    1   "Looking for Quo Vadis at my local video store, I found this 1985 version that looked interesting. Wow! It was amazing! Very much a Ken Ru
"2203_3"    0   "Note to all mad scientists everywhere: if you're going to turn your son into a genetically mutated monster, you need to give him a scarie
"689_1" 0   "What the .......... is this ? This must, without a doubt, be the biggest waste of film, settings and camera ever. I know you can't set your
"9152_1"    0   "Intrigued by the synopsis (every gay video these days has a hunk on the cover; this is not necessarily to be construed as a good sign) I
"6077_1"    0   "Would anyone really watch this RUBBISH if it didn't contain little children running around nude? From a cinematic point of view it is prc
"4656_4"    0   "Unremarkable and unmemorable remake of an old, celebrated English film. Although it may be overly maligned as a total disaster (which it
"9727_7"    1   "Simon Pegg plays a rude crude and often out of control celebrity journalist who is brought from England to work for a big American magazi
"1297_8"    1   "Faithful adaptation of witty and interesting French novel about a cynical and depressed middle-aged software engineer (or something), rel
"5586_8"    1   "Eva (Hedy Lamarr) has just got married with an older man and in the honeymoon, she realizes that her husband does not desire her. Her dis
```

Normal text file
length : 33,556,378   lines : 25,002
Ln : 19   Col : 31   Sel : 0 | 0
Unix (LF)
UTF-8
INS

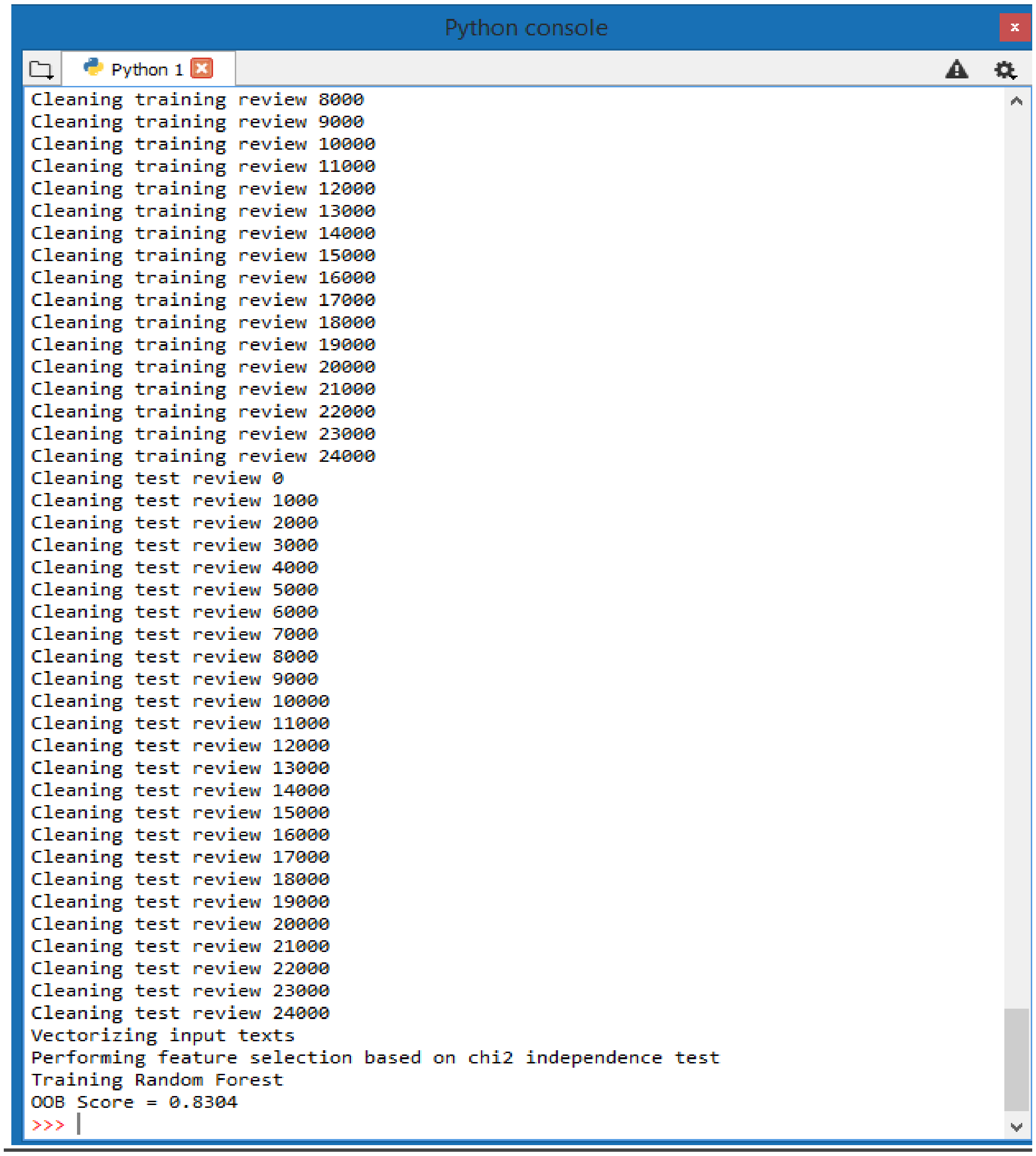
Fig 1.7 Labeled Train Data


C:\Users\lenovo\Documents\sentiment.analysis-master\sentiment.analysis-master\testData.tsv - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
labeledTrainData.tsv
testData.tsv
unlabeledTrainData.tsv

```
id  review
"12311_10"  "Naturally in a film who's main themes are of mortality, nostalgia, and loss of innocence it is perhaps not surprising that it is rated more h
"8348_2"    "This movie is a disaster within a disaster film. It is full of great action scenes, which are only meaningful if you throw away all sense of
"5828_4"    "All in all, this is a movie for kids. We saw it tonight and my child loved it. At one point my kid's excitement was so great that sitting was
"7186_2"    "Afraid of the Dark left me with the impression that several different screenplays were written, all too short for a feature length film, ther
"12128_7"   "A very accurate depiction of small time mob life filmed in New Jersey. The story, characters and script are believable but the acting drops t
"2913_8"    "...as valuable as King Tut's tomb! (OK, maybe not THAT valuable, but worth hunting down if you can). I notice no one has commented on this mo
"4396_1"    "This has to be one of the biggest misfires ever...the script was nice and could have ended a lot better.the actors should have played better
"395_2" "This is one of those movies I watched, and wondered, why did I watch it? What did I find so interesting about it? Being a truck driver myself, I
"10616_1"   "The worst movie i've seen in years (and i've seen a lot of movies). Acting is terrible, there is no plot whatsoever, there is no point whatsc
"9074_9"    "Five medical students (Kevin Bacon, David Labraccio; William Baldwin, Dr. Joe Hurley; Oliver Platt, Randy Steckle; Julia Roberts, Dr. Rachel
"9252_3"    "'The Mill on the Floss' was one of the lesser novels by Mary Ann Evans, who wrote under the male pseudonym George Eliot. I tried to read this
"9896_9"    "I just saw this film at the phoenix film festival today and loved it. The synopsis was listed in our program as \"an old Shakespearean actor
"574_4" "\"The Love Letter\" is one of those movies that could have been really clever, but they wasted it. Focusing on a letter wreaking havoc in a small
"11182_8"   "Another fantastic offering from the Monkey Island team and though it was a long time coming and had to survive the departure of Ron Gilbert i
"11656_4"   "This was included on the disk \"Shorts: Volume 2\"--a rather dull collection of short films. Shorts are among my favorite style of films but
"2322_4"    "I'm not really much of an Abbott & Costello fan (although I do enjoy \"Who's On First\") and, to be honest, there wasn't much in this movie t
"8703_1"    "This movie was dreadful. Biblically very inaccurate. Moses was 80 years old when he led the people out of Egypt, the movie has him about fort
"7483_1"    "I don't think I've ever gave something a 1/10 rating, but this one easily gets the denomination. I find it hard just to sit through one of hi
"6007_10"   "Excellent story-telling and cinematography. Poignant, biting social commentary.<br /><br />Superb effects. Well-filmed and acted.<br /><br />
"12424_4"   "I completely forgot that I'd seen this within a couple of days, which is pretty revealing in itself. The umpteenth version of Gaston 'Phantom
"4672_1"    "I like action movies. I have a softspot for \"B\" flicks with bad dialogue and wooden acting. So, I've been wracking my brain to come up with
"10841_3"   "This is one of the worst Sandra Bullock movie since Speed 2 But not quite that bad. I really lost it with those out of the blue not so \"spec
"8954_7"    "Watched this flick on Saturday afternoon cable. Man, did it drag. I got the metaphors, symbolism, and all that stuff. No, I didn't care one w
"7392_1"    "I went to see \"TKIA\" with high expectations, which might have influence on my opinion on it. I have seen all of the Dogme films, and this T
"10288_8"   "All credit to writer/director Gilles Mimouni who fashioned this winding, twisting tale of deceit and betrayal. While keeping the utmost contr
"5343_4"    "As a writing teacher, there are two ending I never allow my students to use: \"Then I woke up\" and \"Then I Got Run Over by a Truck.\" I am
"4950_1"    "I don't know why this has gotten any decent reviews as it could be the weakest horror comedy I've ever seen. Englund is just in it for a came
"9257_4"    "This film was released in the UK under the name Blood Rites. It was banned outright and never submitted again for release.<br /><br />As The
"8689_3"    "Uncle Fred Olen Ray once again gives us a little of his Retromedia \"goodness\" in the form of this soft-core Cinemax non-classic.<br /><br /
"4480_2"    "OK, it's watchable if you are sick in bed or have nothing else to do. The suspension of disbelief required to get through this movie is signi
"9286_10"   "If I was only allowed to watch one program in my entire life, I would definitely have to pick \"The Chaser's War on Everything\". Of all the
"12245_10"  "A truly terrific, touching film. Female melodrama at its finest, with a lot of comedy: great dialogue, characters and writing. Any woman can
"7529_7"    "Pepe Le Pew can either really creep you out or totally sweep you off your feet. Either way, you can't help feeling a little awe on beholding
"4 4"   "Alas, another Costner movie that was an hour too long. Credible performances, but the script had no where to go and was in no hurry to get there.
```

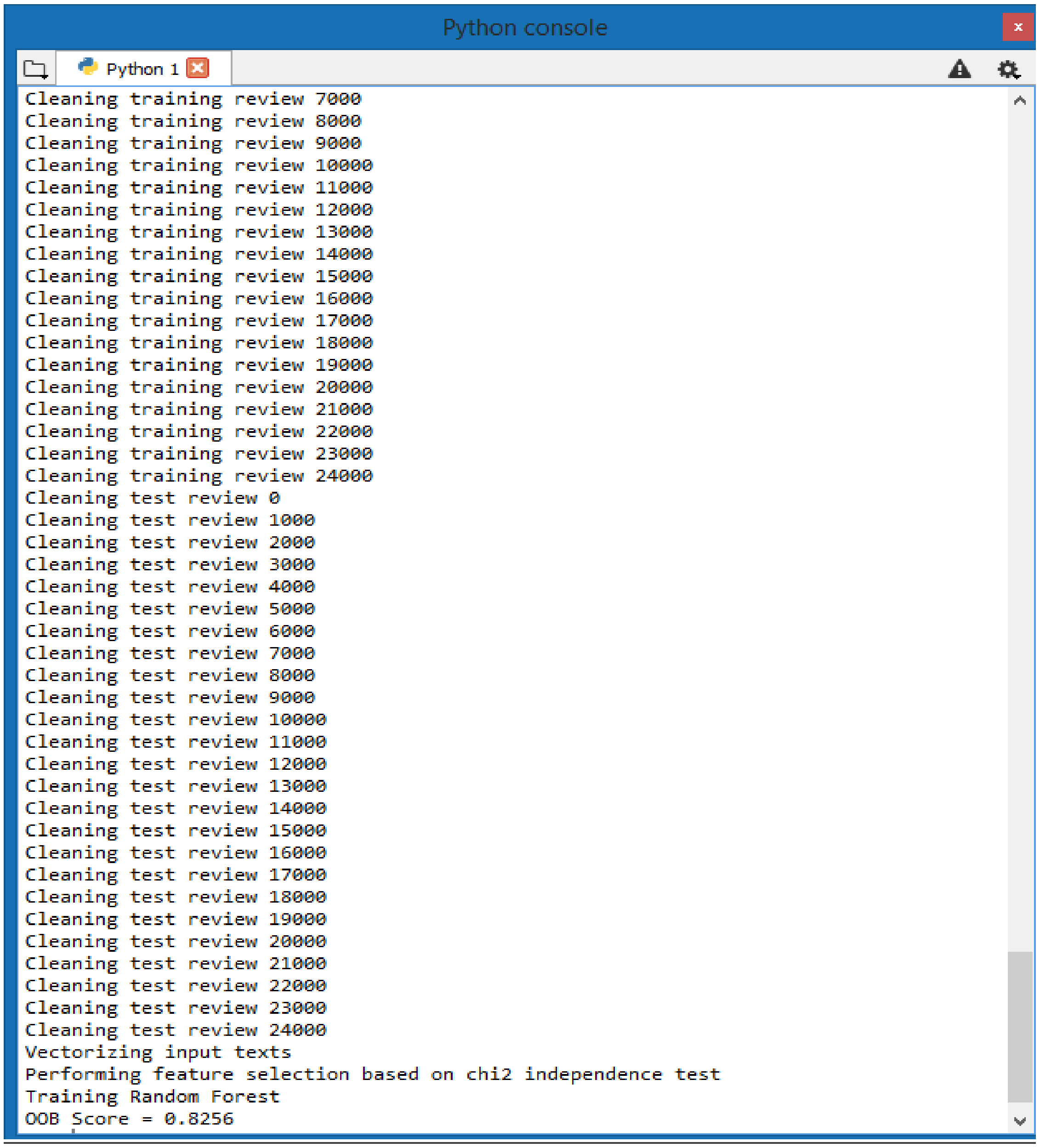
```
Normal text file    length : 32,724,746    lines : 25,002    Ln : 1    Col : 1    Sel : 0 | 0    Unix (LF)    UTF-8    INS
```

Fig 1.8 Test Data



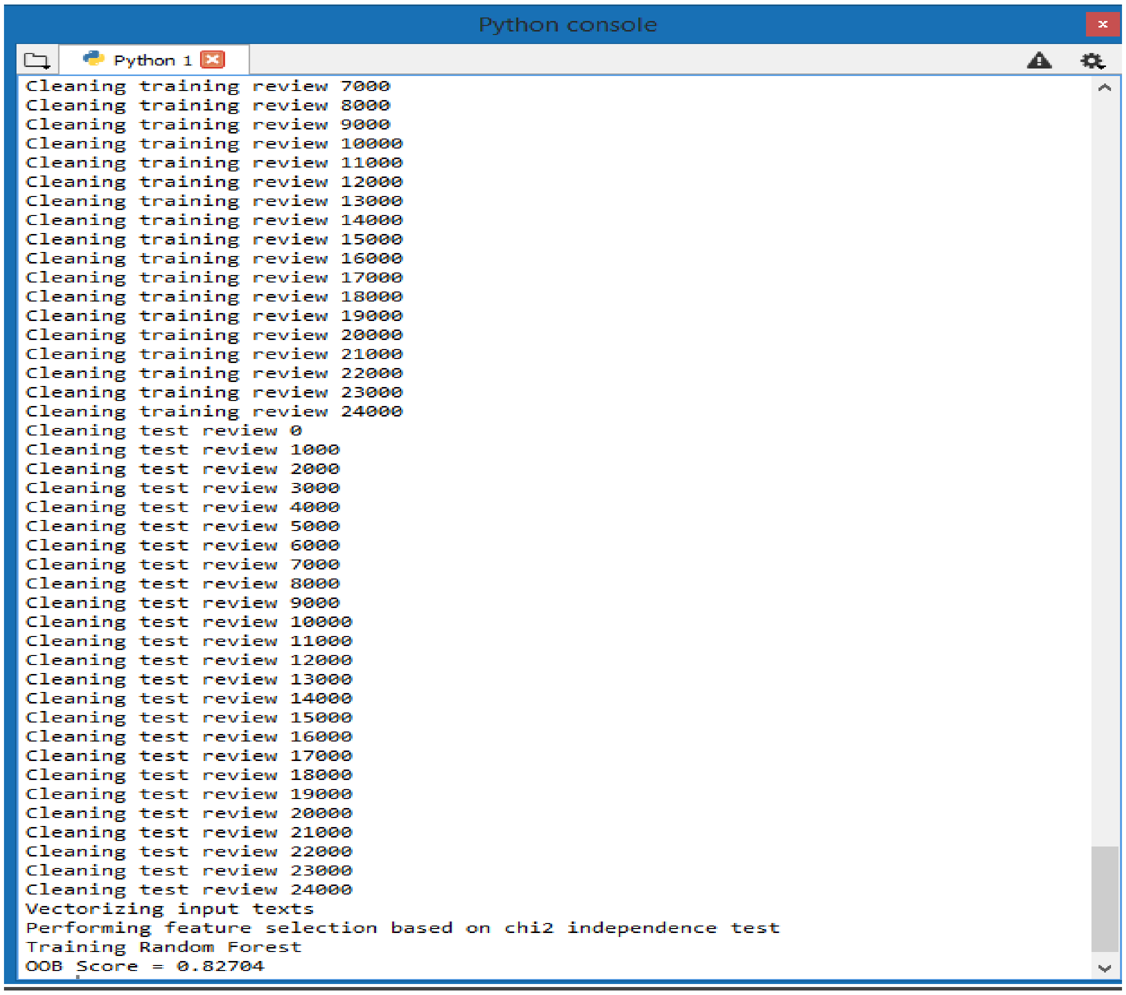
```
Cleaning training review 8000
Cleaning training review 9000
Cleaning training review 10000
Cleaning training review 11000
Cleaning training review 12000
Cleaning training review 13000
Cleaning training review 14000
Cleaning training review 15000
Cleaning training review 16000
Cleaning training review 17000
Cleaning training review 18000
Cleaning training review 19000
Cleaning training review 20000
Cleaning training review 21000
Cleaning training review 22000
Cleaning training review 23000
Cleaning training review 24000
Cleaning test review 0
Cleaning test review 1000
Cleaning test review 2000
Cleaning test review 3000
Cleaning test review 4000
Cleaning test review 5000
Cleaning test review 6000
Cleaning test review 7000
Cleaning test review 8000
Cleaning test review 9000
Cleaning test review 10000
Cleaning test review 11000
Cleaning test review 12000
Cleaning test review 13000
Cleaning test review 14000
Cleaning test review 15000
Cleaning test review 16000
Cleaning test review 17000
Cleaning test review 18000
Cleaning test review 19000
Cleaning test review 20000
Cleaning test review 21000
Cleaning test review 22000
Cleaning test review 23000
Cleaning test review 24000
Vectorizing input texts
Performing feature selection based on chi2 independence test
Training Random Forest
OOB Score = 0.8304
>>> 
```

<u>Fig 1.9 Random Forest TFIDF</u>



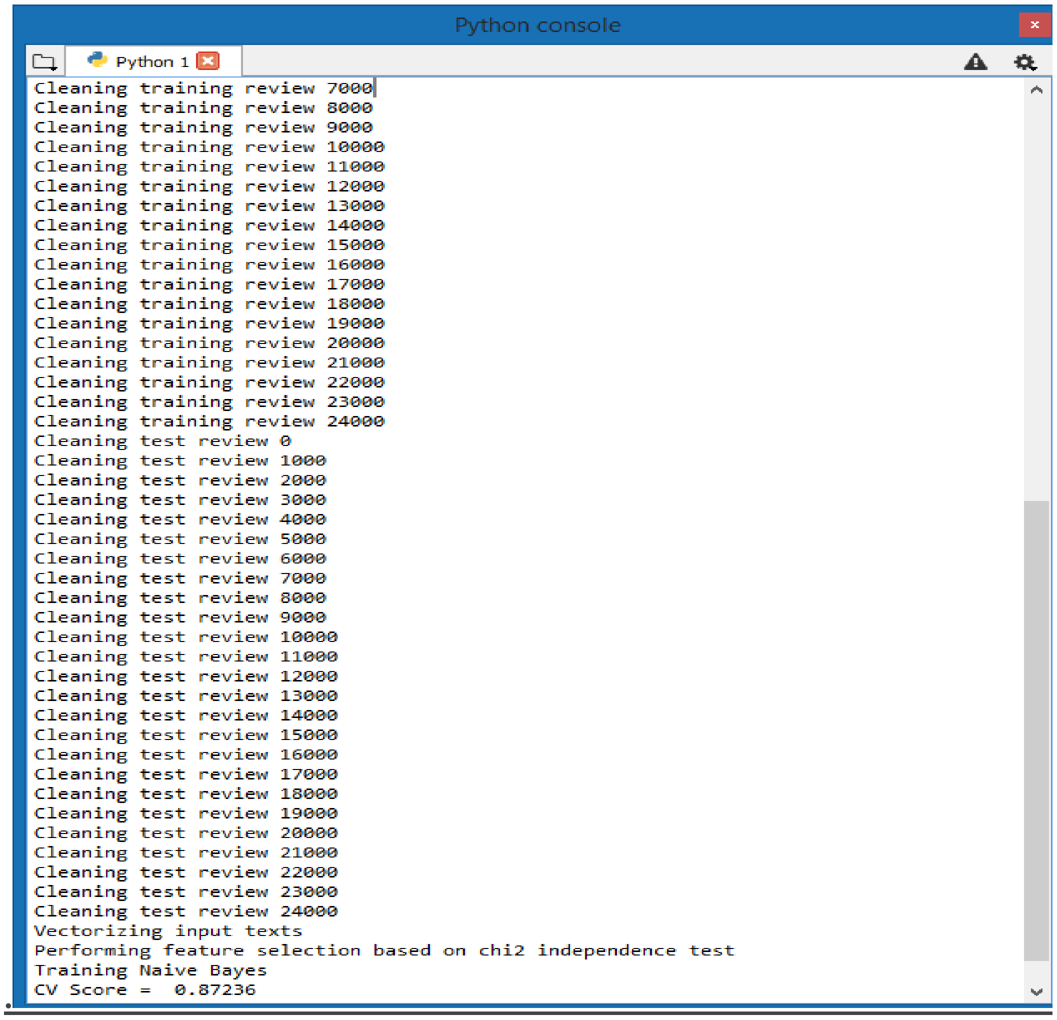
```
Cleaning training review 7000
Cleaning training review 8000
Cleaning training review 9000
Cleaning training review 10000
Cleaning training review 11000
Cleaning training review 12000
Cleaning training review 13000
Cleaning training review 14000
Cleaning training review 15000
Cleaning training review 16000
Cleaning training review 17000
Cleaning training review 18000
Cleaning training review 19000
Cleaning training review 20000
Cleaning training review 21000
Cleaning training review 22000
Cleaning training review 23000
Cleaning training review 24000
Cleaning test review 0
Cleaning test review 1000
Cleaning test review 2000
Cleaning test review 3000
Cleaning test review 4000
Cleaning test review 5000
Cleaning test review 6000
Cleaning test review 7000
Cleaning test review 8000
Cleaning test review 9000
Cleaning test review 10000
Cleaning test review 11000
Cleaning test review 12000
Cleaning test review 13000
Cleaning test review 14000
Cleaning test review 15000
Cleaning test review 16000
Cleaning test review 17000
Cleaning test review 18000
Cleaning test review 19000
Cleaning test review 20000
Cleaning test review 21000
Cleaning test review 22000
Cleaning test review 23000
Cleaning test review 24000
Vectorizing input texts
Performing feature selection based on chi2 independence test
Training Random Forest
OOB Score = 0.8256
```

Fig 1.10 Random Forest-Int



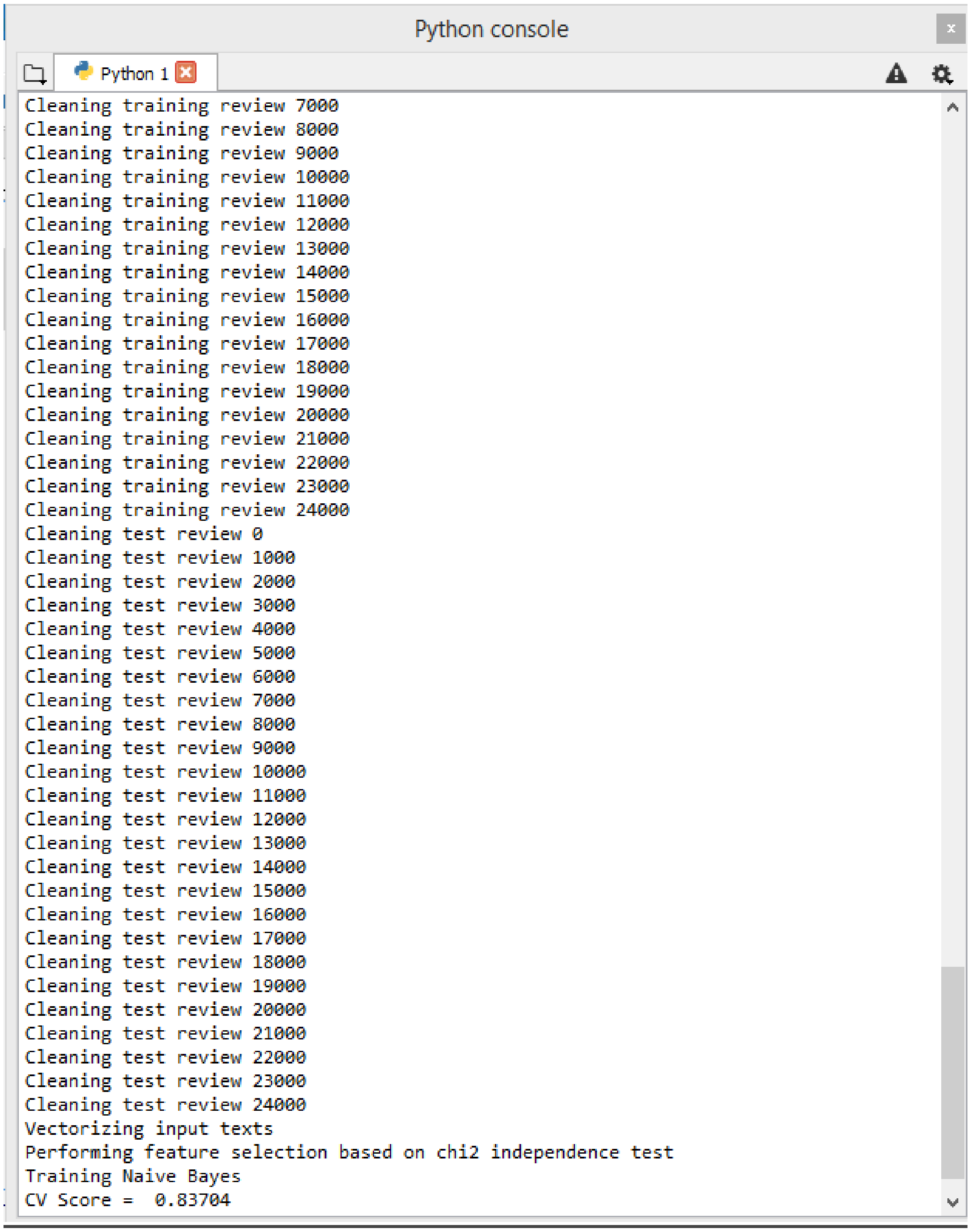
```
Cleaning training review 7000
Cleaning training review 8000
Cleaning training review 9000
Cleaning training review 10000
Cleaning training review 11000
Cleaning training review 12000
Cleaning training review 13000
Cleaning training review 14000
Cleaning training review 15000
Cleaning training review 16000
Cleaning training review 17000
Cleaning training review 18000
Cleaning training review 19000
Cleaning training review 20000
Cleaning training review 21000
Cleaning training review 22000
Cleaning training review 23000
Cleaning training review 24000
Cleaning test review 0
Cleaning test review 1000
Cleaning test review 2000
Cleaning test review 3000
Cleaning test review 4000
Cleaning test review 5000
Cleaning test review 6000
Cleaning test review 7000
Cleaning test review 8000
Cleaning test review 9000
Cleaning test review 10000
Cleaning test review 11000
Cleaning test review 12000
Cleaning test review 13000
Cleaning test review 14000
Cleaning test review 15000
Cleaning test review 16000
Cleaning test review 17000
Cleaning test review 18000
Cleaning test review 19000
Cleaning test review 20000
Cleaning test review 21000
Cleaning test review 22000
Cleaning test review 23000
Cleaning test review 24000
Vectorizing input texts
Performing feature selection based on chi2 independence test
Training Random Forest
OOB Score = 0.82704
```

Fig 1.11 Random Forest- Binary



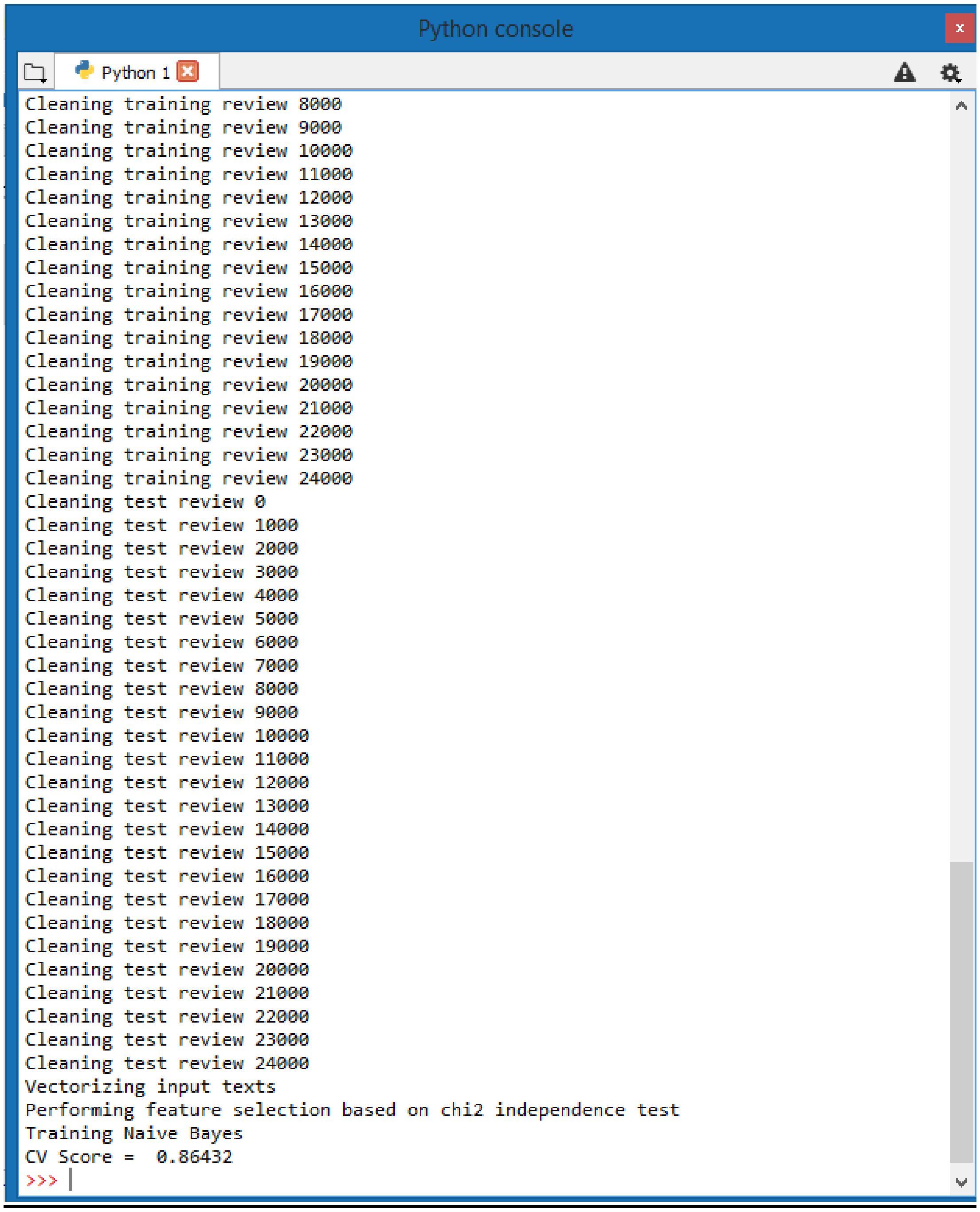
```
Cleaning training review 7000
Cleaning training review 8000
Cleaning training review 9000
Cleaning training review 10000
Cleaning training review 11000
Cleaning training review 12000
Cleaning training review 13000
Cleaning training review 14000
Cleaning training review 15000
Cleaning training review 16000
Cleaning training review 17000
Cleaning training review 18000
Cleaning training review 19000
Cleaning training review 20000
Cleaning training review 21000
Cleaning training review 22000
Cleaning training review 23000
Cleaning training review 24000
Cleaning test review 0
Cleaning test review 1000
Cleaning test review 2000
Cleaning test review 3000
Cleaning test review 4000
Cleaning test review 5000
Cleaning test review 6000
Cleaning test review 7000
Cleaning test review 8000
Cleaning test review 9000
Cleaning test review 10000
Cleaning test review 11000
Cleaning test review 12000
Cleaning test review 13000
Cleaning test review 14000
Cleaning test review 15000
Cleaning test review 16000
Cleaning test review 17000
Cleaning test review 18000
Cleaning test review 19000
Cleaning test review 20000
Cleaning test review 21000
Cleaning test review 22000
Cleaning test review 23000
Cleaning test review 24000
Vectorizing input texts
Performing feature selection based on chi2 independence test
Training Naive Bayes
CV Score =  0.87236
```

Fig 1.12 Naive Bayes – TFIDF



```
Cleaning training review 7000
Cleaning training review 8000
Cleaning training review 9000
Cleaning training review 10000
Cleaning training review 11000
Cleaning training review 12000
Cleaning training review 13000
Cleaning training review 14000
Cleaning training review 15000
Cleaning training review 16000
Cleaning training review 17000
Cleaning training review 18000
Cleaning training review 19000
Cleaning training review 20000
Cleaning training review 21000
Cleaning training review 22000
Cleaning training review 23000
Cleaning training review 24000
Cleaning test review 0
Cleaning test review 1000
Cleaning test review 2000
Cleaning test review 3000
Cleaning test review 4000
Cleaning test review 5000
Cleaning test review 6000
Cleaning test review 7000
Cleaning test review 8000
Cleaning test review 9000
Cleaning test review 10000
Cleaning test review 11000
Cleaning test review 12000
Cleaning test review 13000
Cleaning test review 14000
Cleaning test review 15000
Cleaning test review 16000
Cleaning test review 17000
Cleaning test review 18000
Cleaning test review 19000
Cleaning test review 20000
Cleaning test review 21000
Cleaning test review 22000
Cleaning test review 23000
Cleaning test review 24000
Vectorizing input texts
Performing feature selection based on chi2 independence test
Training Naive Bayes
CV Score =  0.83704
```

Fig 1.13 Naive Bayes – Int

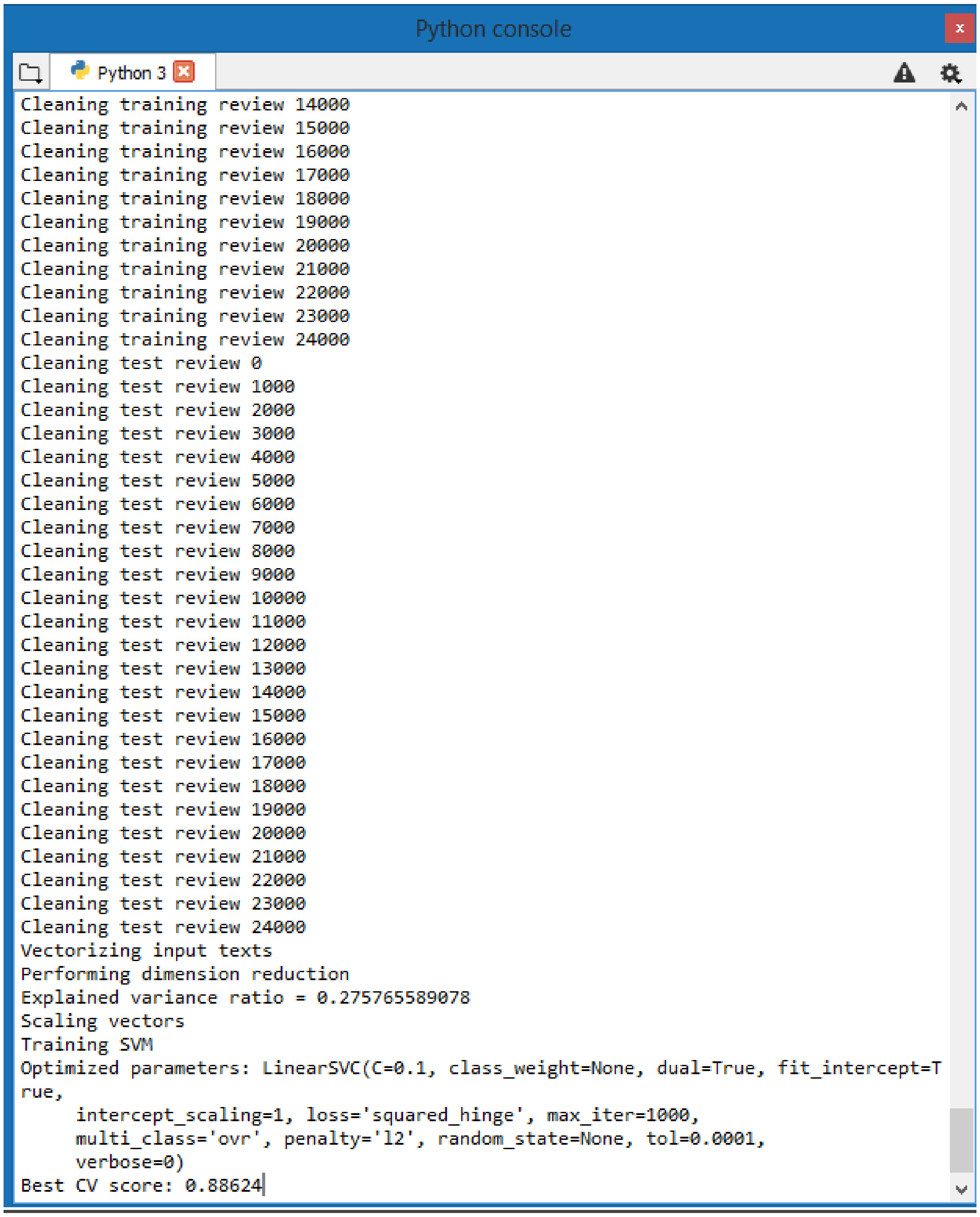


Python console

Python 1

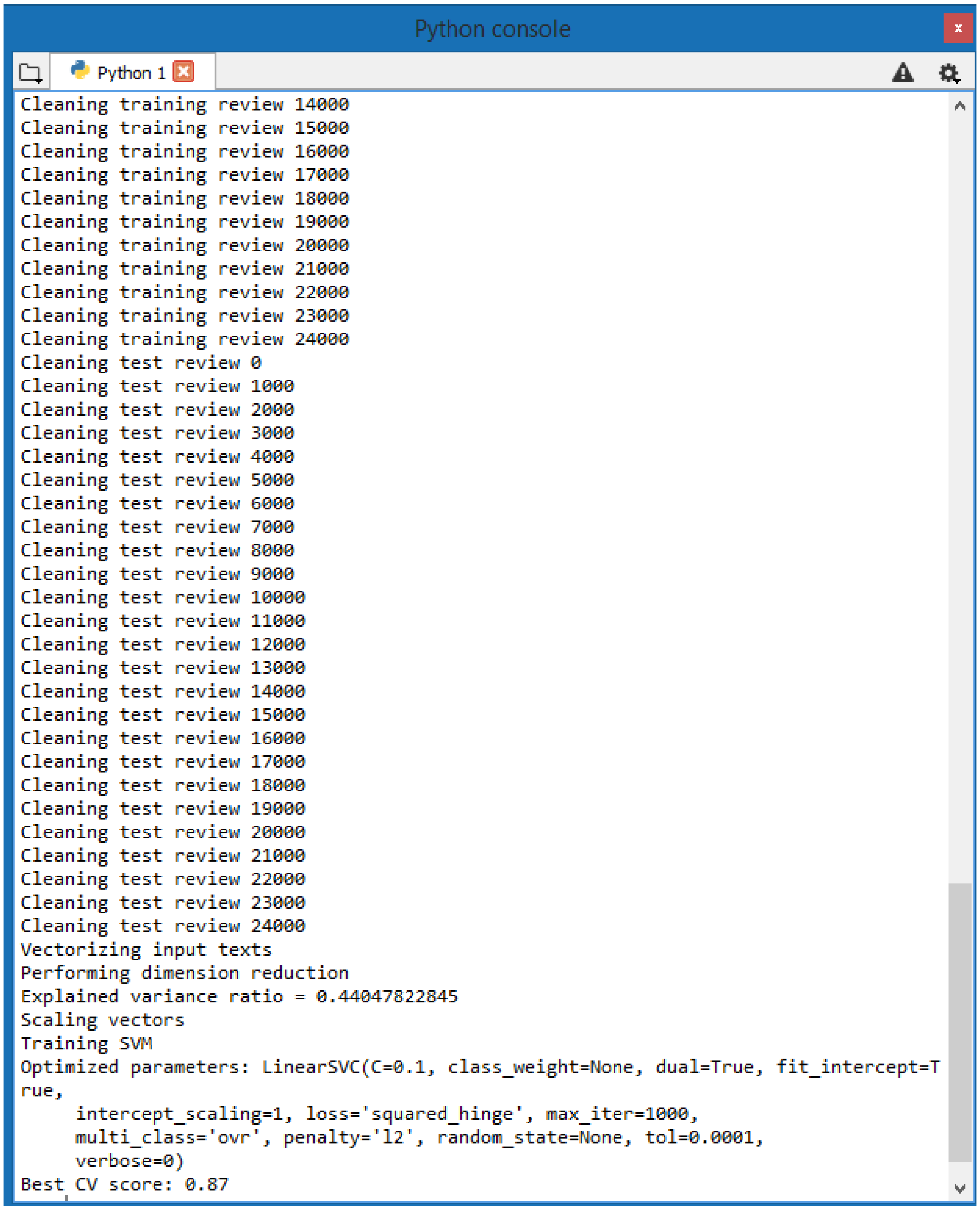
```
Cleaning training review 8000
Cleaning training review 9000
Cleaning training review 10000
Cleaning training review 11000
Cleaning training review 12000
Cleaning training review 13000
Cleaning training review 14000
Cleaning training review 15000
Cleaning training review 16000
Cleaning training review 17000
Cleaning training review 18000
Cleaning training review 19000
Cleaning training review 20000
Cleaning training review 21000
Cleaning training review 22000
Cleaning training review 23000
Cleaning training review 24000
Cleaning test review 0
Cleaning test review 1000
Cleaning test review 2000
Cleaning test review 3000
Cleaning test review 4000
Cleaning test review 5000
Cleaning test review 6000
Cleaning test review 7000
Cleaning test review 8000
Cleaning test review 9000
Cleaning test review 10000
Cleaning test review 11000
Cleaning test review 12000
Cleaning test review 13000
Cleaning test review 14000
Cleaning test review 15000
Cleaning test review 16000
Cleaning test review 17000
Cleaning test review 18000
Cleaning test review 19000
Cleaning test review 20000
Cleaning test review 21000
Cleaning test review 22000
Cleaning test review 23000
Cleaning test review 24000
Vectorizing input texts
Performing feature selection based on chi2 independence test
Training Naive Bayes
CV Score =  0.86432
>>> 
```

Fig 1.14 Naïve-Bayes Binary



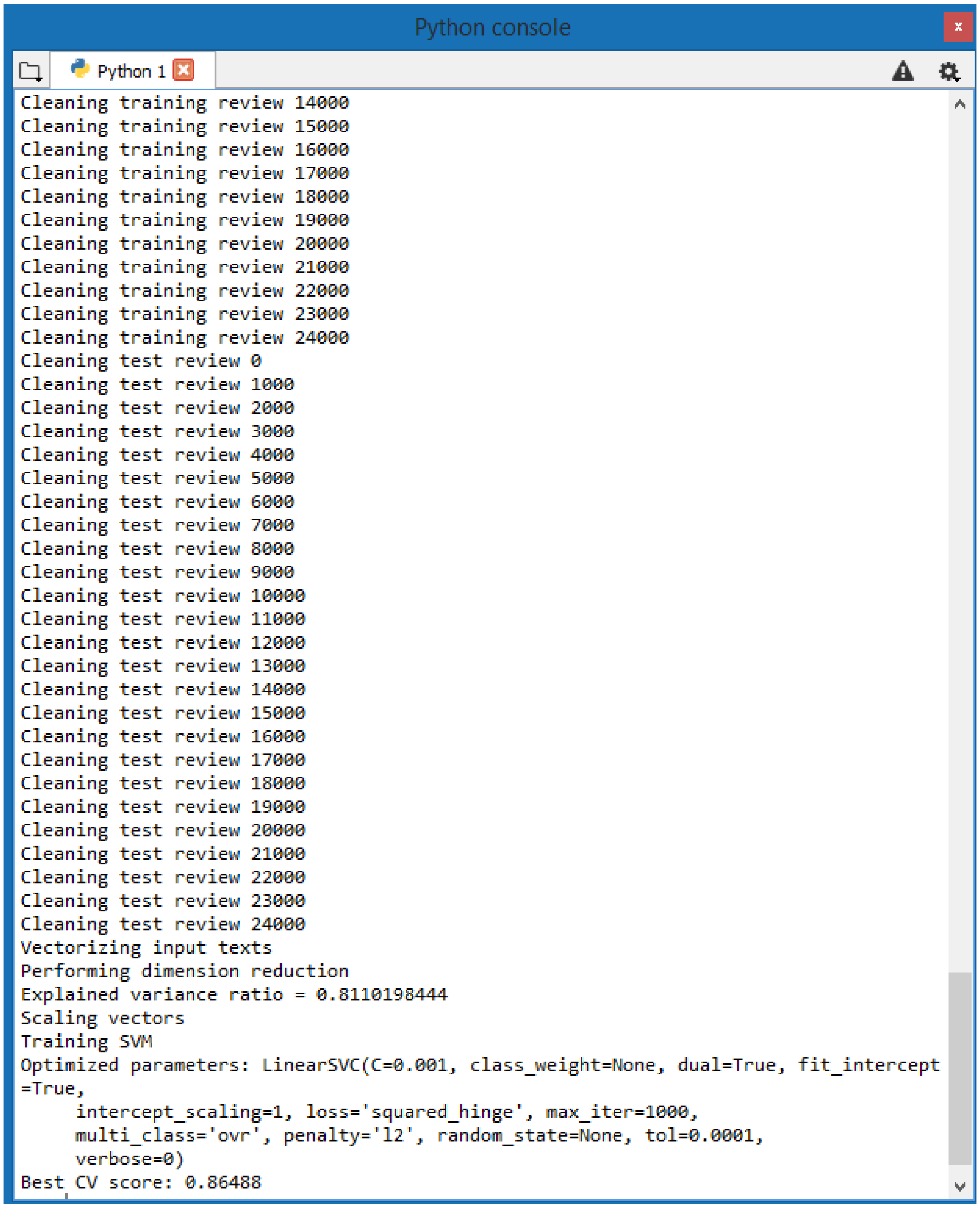
```
Cleaning training review 14000
Cleaning training review 15000
Cleaning training review 16000
Cleaning training review 17000
Cleaning training review 18000
Cleaning training review 19000
Cleaning training review 20000
Cleaning training review 21000
Cleaning training review 22000
Cleaning training review 23000
Cleaning training review 24000
Cleaning test review 0
Cleaning test review 1000
Cleaning test review 2000
Cleaning test review 3000
Cleaning test review 4000
Cleaning test review 5000
Cleaning test review 6000
Cleaning test review 7000
Cleaning test review 8000
Cleaning test review 9000
Cleaning test review 10000
Cleaning test review 11000
Cleaning test review 12000
Cleaning test review 13000
Cleaning test review 14000
Cleaning test review 15000
Cleaning test review 16000
Cleaning test review 17000
Cleaning test review 18000
Cleaning test review 19000
Cleaning test review 20000
Cleaning test review 21000
Cleaning test review 22000
Cleaning test review 23000
Cleaning test review 24000
Vectorizing input texts
Performing dimension reduction
Explained variance ratio = 0.275765589078
Scaling vectors
Training SVM
Optimized parameters: LinearSVC(C=0.1, class_weight=None, dual=True, fit_intercept=T
rue,
     intercept_scaling=1, loss='squared_hinge', max_iter=1000,
     multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
     verbose=0)
Best CV score: 0.88624
```

Fig 1.15SVM- TFIDF



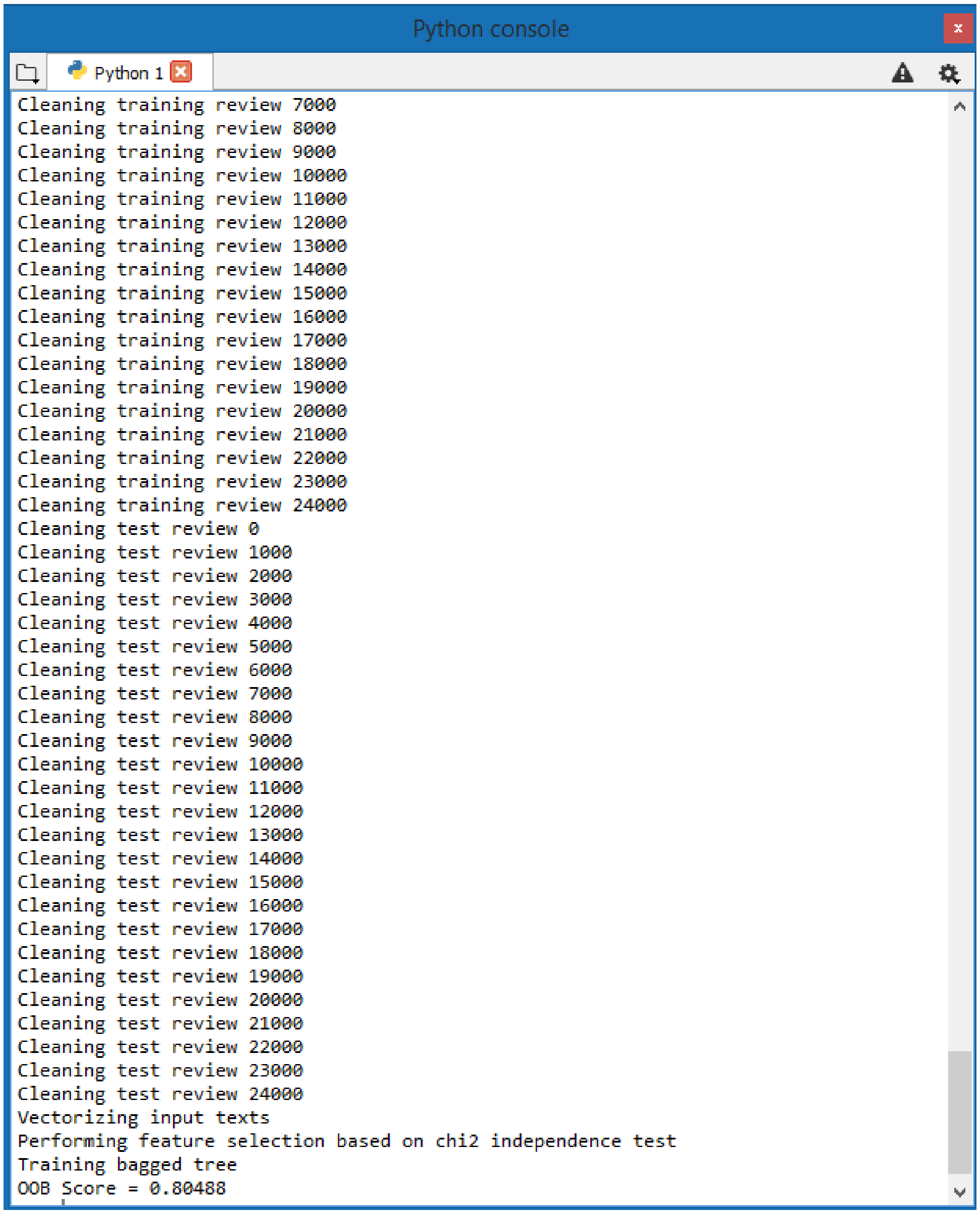
```
Cleaning training review 14000
Cleaning training review 15000
Cleaning training review 16000
Cleaning training review 17000
Cleaning training review 18000
Cleaning training review 19000
Cleaning training review 20000
Cleaning training review 21000
Cleaning training review 22000
Cleaning training review 23000
Cleaning training review 24000
Cleaning test review 0
Cleaning test review 1000
Cleaning test review 2000
Cleaning test review 3000
Cleaning test review 4000
Cleaning test review 5000
Cleaning test review 6000
Cleaning test review 7000
Cleaning test review 8000
Cleaning test review 9000
Cleaning test review 10000
Cleaning test review 11000
Cleaning test review 12000
Cleaning test review 13000
Cleaning test review 14000
Cleaning test review 15000
Cleaning test review 16000
Cleaning test review 17000
Cleaning test review 18000
Cleaning test review 19000
Cleaning test review 20000
Cleaning test review 21000
Cleaning test review 22000
Cleaning test review 23000
Cleaning test review 24000
Vectorizing input texts
Performing dimension reduction
Explained variance ratio = 0.44047822845
Scaling vectors
Training SVM
Optimized parameters: LinearSVC(C=0.1, class_weight=None, dual=True, fit_intercept=T
rue,
     intercept_scaling=1, loss='squared_hinge', max_iter=1000,
     multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
     verbose=0)
Best CV score: 0.87
```

Fig 1.16 SVM-Binary



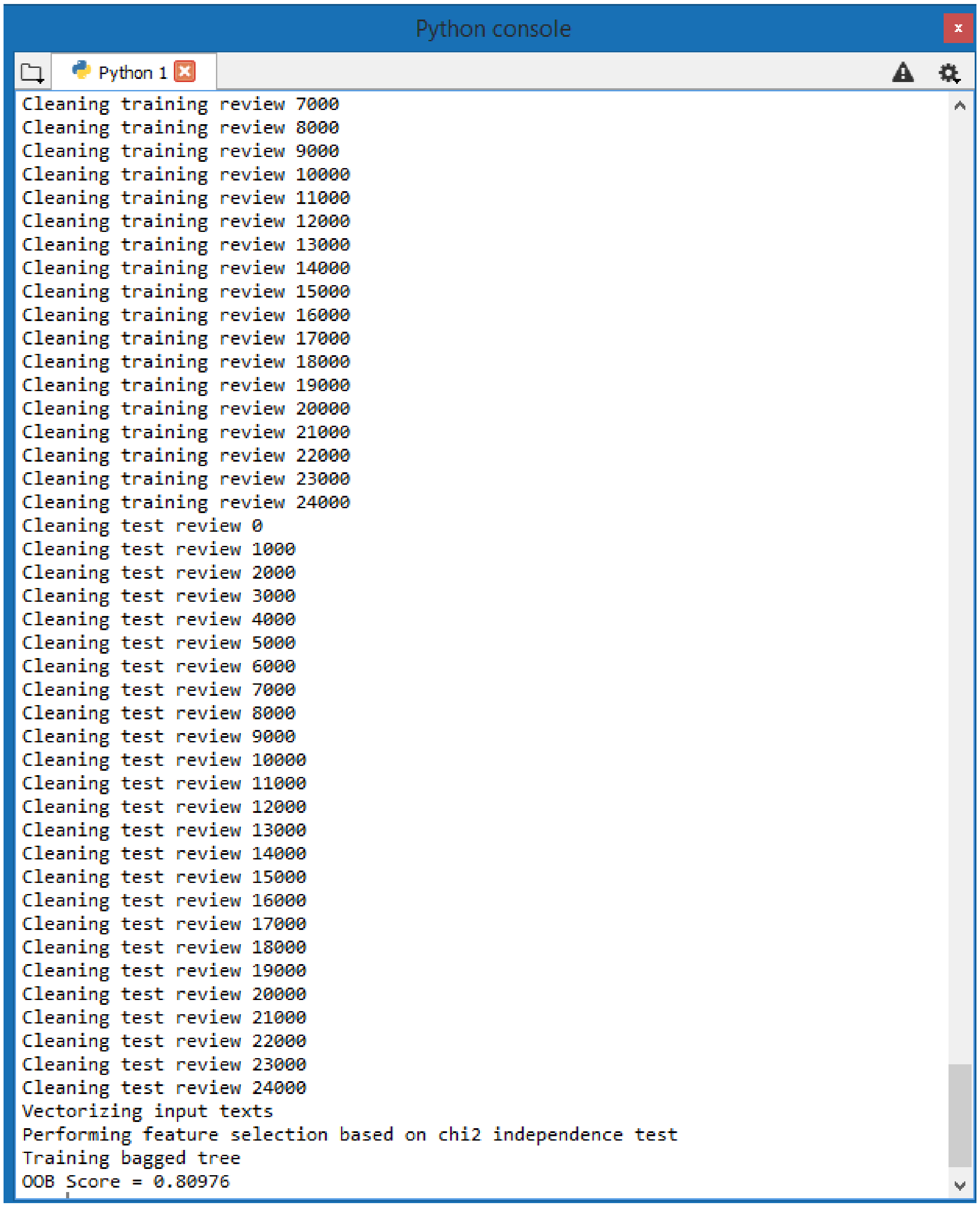
```
Cleaning training review 14000
Cleaning training review 15000
Cleaning training review 16000
Cleaning training review 17000
Cleaning training review 18000
Cleaning training review 19000
Cleaning training review 20000
Cleaning training review 21000
Cleaning training review 22000
Cleaning training review 23000
Cleaning training review 24000
Cleaning test review 0
Cleaning test review 1000
Cleaning test review 2000
Cleaning test review 3000
Cleaning test review 4000
Cleaning test review 5000
Cleaning test review 6000
Cleaning test review 7000
Cleaning test review 8000
Cleaning test review 9000
Cleaning test review 10000
Cleaning test review 11000
Cleaning test review 12000
Cleaning test review 13000
Cleaning test review 14000
Cleaning test review 15000
Cleaning test review 16000
Cleaning test review 17000
Cleaning test review 18000
Cleaning test review 19000
Cleaning test review 20000
Cleaning test review 21000
Cleaning test review 22000
Cleaning test review 23000
Cleaning test review 24000
Vectorizing input texts
Performing dimension reduction
Explained variance ratio = 0.8110198444
Scaling vectors
Training SVM
Optimized parameters: LinearSVC(C=0.001, class_weight=None, dual=True, fit_intercept
=True,
     intercept_scaling=1, loss='squared_hinge', max_iter=1000,
     multi_class='ovr', penalty='l2', random_state=None, tol=0.0001,
     verbose=0)
Best CV score: 0.86488
```

Fig 1.17 SVM-Int



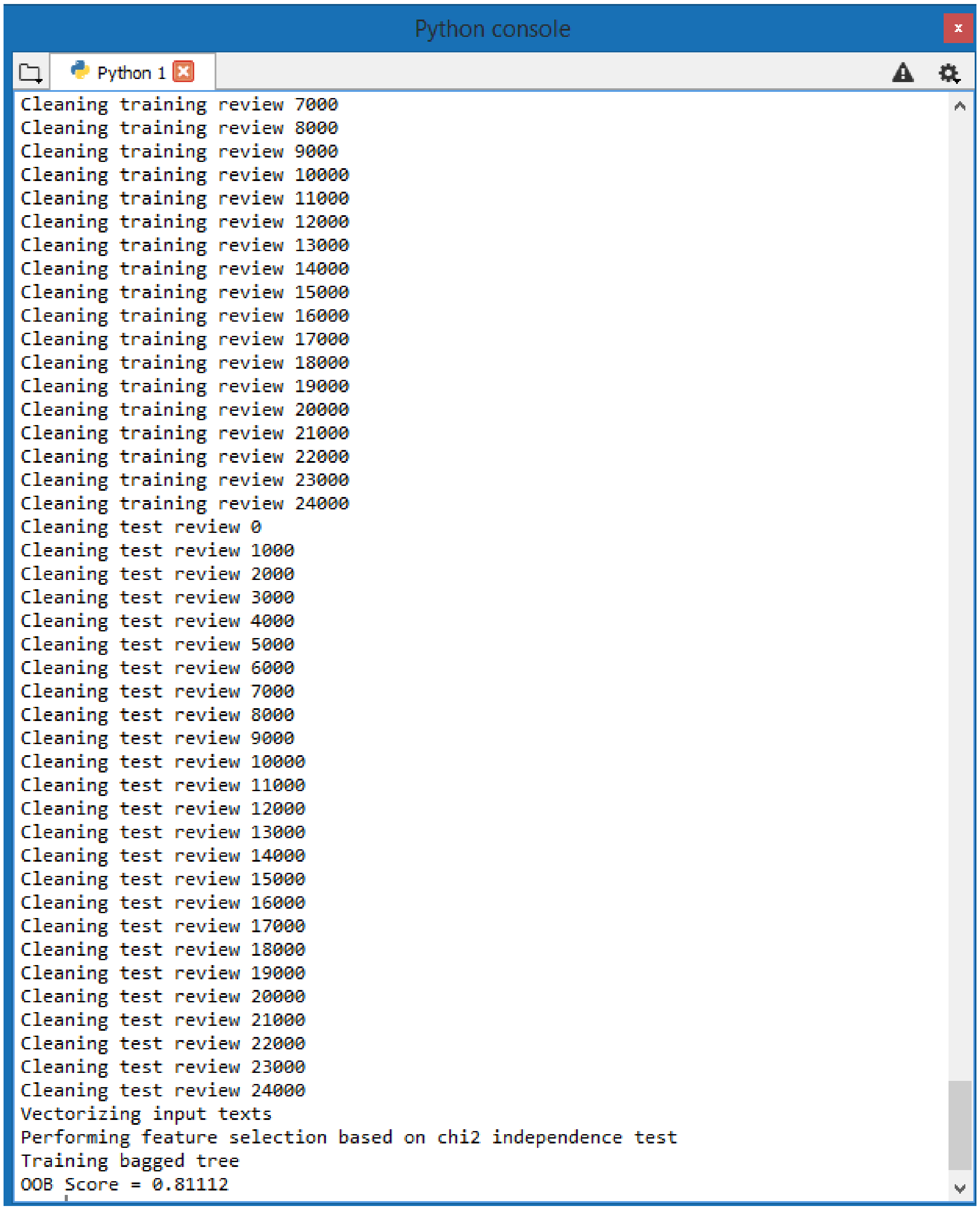
```
Cleaning training review 7000
Cleaning training review 8000
Cleaning training review 9000
Cleaning training review 10000
Cleaning training review 11000
Cleaning training review 12000
Cleaning training review 13000
Cleaning training review 14000
Cleaning training review 15000
Cleaning training review 16000
Cleaning training review 17000
Cleaning training review 18000
Cleaning training review 19000
Cleaning training review 20000
Cleaning training review 21000
Cleaning training review 22000
Cleaning training review 23000
Cleaning training review 24000
Cleaning test review 0
Cleaning test review 1000
Cleaning test review 2000
Cleaning test review 3000
Cleaning test review 4000
Cleaning test review 5000
Cleaning test review 6000
Cleaning test review 7000
Cleaning test review 8000
Cleaning test review 9000
Cleaning test review 10000
Cleaning test review 11000
Cleaning test review 12000
Cleaning test review 13000
Cleaning test review 14000
Cleaning test review 15000
Cleaning test review 16000
Cleaning test review 17000
Cleaning test review 18000
Cleaning test review 19000
Cleaning test review 20000
Cleaning test review 21000
Cleaning test review 22000
Cleaning test review 23000
Cleaning test review 24000
Vectorizing input texts
Performing feature selection based on chi2 independence test
Training bagged tree
OOB Score = 0.80488
```

Fig 1.18 Bagged Tree- Int



```
Cleaning training review 7000
Cleaning training review 8000
Cleaning training review 9000
Cleaning training review 10000
Cleaning training review 11000
Cleaning training review 12000
Cleaning training review 13000
Cleaning training review 14000
Cleaning training review 15000
Cleaning training review 16000
Cleaning training review 17000
Cleaning training review 18000
Cleaning training review 19000
Cleaning training review 20000
Cleaning training review 21000
Cleaning training review 22000
Cleaning training review 23000
Cleaning training review 24000
Cleaning test review 0
Cleaning test review 1000
Cleaning test review 2000
Cleaning test review 3000
Cleaning test review 4000
Cleaning test review 5000
Cleaning test review 6000
Cleaning test review 7000
Cleaning test review 8000
Cleaning test review 9000
Cleaning test review 10000
Cleaning test review 11000
Cleaning test review 12000
Cleaning test review 13000
Cleaning test review 14000
Cleaning test review 15000
Cleaning test review 16000
Cleaning test review 17000
Cleaning test review 18000
Cleaning test review 19000
Cleaning test review 20000
Cleaning test review 21000
Cleaning test review 22000
Cleaning test review 23000
Cleaning test review 24000
Vectorizing input texts
Performing feature selection based on chi2 independence test
Training bagged tree
OOB Score = 0.80976
```

<u>**Fig 1.19 Bagged Tree- Binary**</u>



```
Cleaning training review 7000
Cleaning training review 8000
Cleaning training review 9000
Cleaning training review 10000
Cleaning training review 11000
Cleaning training review 12000
Cleaning training review 13000
Cleaning training review 14000
Cleaning training review 15000
Cleaning training review 16000
Cleaning training review 17000
Cleaning training review 18000
Cleaning training review 19000
Cleaning training review 20000
Cleaning training review 21000
Cleaning training review 22000
Cleaning training review 23000
Cleaning training review 24000
Cleaning test review 0
Cleaning test review 1000
Cleaning test review 2000
Cleaning test review 3000
Cleaning test review 4000
Cleaning test review 5000
Cleaning test review 6000
Cleaning test review 7000
Cleaning test review 8000
Cleaning test review 9000
Cleaning test review 10000
Cleaning test review 11000
Cleaning test review 12000
Cleaning test review 13000
Cleaning test review 14000
Cleaning test review 15000
Cleaning test review 16000
Cleaning test review 17000
Cleaning test review 18000
Cleaning test review 19000
Cleaning test review 20000
Cleaning test review 21000
Cleaning test review 22000
Cleaning test review 23000
Cleaning test review 24000
Vectorizing input texts
Performing feature selection based on chi2 independence test
Training bagged tree
OOB Score = 0.81112
```

Fig 1.20 Bagged Tree-TFIDF

Chapter-7

Conclusion and Future Work

In this work, we studied a wide range of NLP classification models. Our investigations consisted of two main parts. In the first part, we used the dataset provided by Kaggle and applied the bag of words, and word2vec models and Tf-idf to represent words numerically. We then used several classifiers, including random forest, SVM, Naive Bayes and Bagged tree to perform the binary classification task. When we use these classifiers, one of the challenges is to aggregate word vectors into a single feature vector for each review. We tried vector averaging, and clustering to produce the aggregated feature vectors. Then each classifier which we have used produced a cv score as an output in result, from there we have compared the results of each classifier on basis of its cv score.

From the performance table of cv scores of different classifiers, we can conclude that Random Forest does not suit well in text data, since random forest makes the split based on randomly selected feature, while text data are sparse vectors and we may select and split based on irrelevant features.

For the given text data, we can observe that linear SVM (Support Vector Machine) does the best job of predicting the sentiments or opinion of the reviews given by the user. Linear SVM with tf-idf yields the best result with 88.624% accuracy.

There are many possible directions for future work. For example, we would like to explore the effect of using only subjective sentences in classifying reviews for both the term counting methods and the ML method. The positive and negative terms may not all be equally positive or negative. Positive and negative terms can 20 be given weights (those could be, for example, their SO-PMI scores) to show just how positive or negative they are. Overstatements and understatements could also be weighted. Another way to improve the accuracy of classifying movie reviews could be to automatically build small domain models of salient objective key phrases.

Positive and negative terms in these key phrases would be ignored in the term-counting method. In the machine learning experiments, one direction of future research is to identify the named entities and to measure their contribution to the accuracy of the SVM classifiers. For example, it could be the case that some names of popular actors or directors appear most often in movies with positive reviews. User-based reviews are informal text. This means that it is highly possible it contains misspellings, slang, abbreviations and/or emojis. Misspelled adjectives are not detected by our part-of-speech tagger and if we could detect those adjectives, the result would look different.

The system is not trained on slang, abbreviations or emoticons. It would be interesting to see if such a system could improve the results. In 2015 Kralj Novak et al. created their own emoji sentiment lexicon and the detection of emoji is already popular within sentiment analysis of the micro-blogging area.

Chapter-8

Bibliography

[1] P.Kalaivani, Dr.K.L.Shunmuganathan, "Sentiment classification of movie review by supervised machine learning approach", Indian Journal of Computer Science and Engineering (IJCSE) ,Vol. 4 No.4, Aug-Sep 2013.

[2] Suchita V. Wawre , Sachin N. Deshmukh , "Sentimental Analysis of Movie Review using Machine Learning Algorithm with Tuned Hype parameter ", Dept. of Computer Science and Information Technology, Dr B. A. M. University Aurangabad, Maharashtra, India, Vol. 4, Issue 6, June 2016.

[3] Alistair Kennedy and Diana Inkpen, "Sentiment Classification of Movie Reviews Using Contextual Valence Shifters". In Computational Intelligence June-July,2015.

[4] Alejandro Pelaez, Talal Ahmed, Mohsen Ghassemi., "In Sentiment analysis of IMDb movie reviews", Rutgers University, Spring 2015.

[5] Bo Pang and Lillian Lee and Shivakumar Vaithyanathan, "Thumbs up? Sentiment Classification using Machine Learning Techniques", Language Processing (EMNLP), Philadelphia, pp. 79-86, July 2002.

[6] James Hong and Michael Fang, "Sentiment Analysis with Deeply Learned Distributed Representations of Variable Length Text", 2011.

[7] Aina Elisabeth Thunestveit, "Sentiment Analysis on User-Based Reviews: Movie Recommendation Case " ,Norwegian University of Science and Technology, June 2016.

[8] Turing, A.M. (1950). Computing machinery and intelligence. Mind, 59, 433-460.

[9] Andreas C. Muller & Sarah Guido - Introduction to Machine Learning, 2017.

[10] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: Sentiment classification using machine learning techniques,” in Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10, EMNLP ’02, (Stroudsburg, PA, USA), pp. 79–86, Association for Computational Linguistics, 2002.

Online references

1. www.machinelearningmastery.com

2. https://code.google.com/archive/p/word2vec/

3. www.kaggle.com/c/word2vec-nlp-tutorial

4. https://deeplearning4j.org/datavec

5. www.scikit-learn.org

6. http://stackoverflow.com/questions/tagged/machine-learning

7. https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

8. http://en.wikipedia.org/wiki/Machine_learning

9. https://store.continuum.io/cshop/anaconda/