**CHAPTER 1**

**INTRODUCTION**

**1.1 BIG DATA**

Big data refers to data sets that are too large or complex for traditional data-processing application software to adequately deal with. Data with many cases (rows) offer greater statistical power, while data with higher complexity (more attributes or columns) may lead to a higher false discovery rate. Big data challenges include capturing data, data storage, data analysis, search, sharing, transfer, visualization, querying, updating, information privacy and data source.

Current usage of the term big data tends to refer to the use of predictive analytics, user behavior analytics, or certain other advanced data analytics methods that extract value from data, and seldom to a particular size of data set. Analysis of data sets can find new correlations to "spot business trends, prevent diseases, combat crime and so on."Scientists, business executives, practitioners of medicine, advertising and governments alike regularly meet difficulties with large data-sets in areas including Internet search, fintech, urban informatics, and business informatics. Data sets grow rapidly- in part because they are increasingly gathered by cheap and numerous information- sensing Internet of things devices such as mobile devices, aerial (remote sensing), software logs, cameras, microphones, radio-frequency identification (RFID) readers and wireless sensor networks. The world's technological per-capita capacity to store information has roughly doubled every 40 months since the 1980s; as of 2012, every day 2.5 exabytes (2.5×1018) of data are generated. Based on an IDC report prediction, the global data volume will grow exponentially from 4.4 zettabytes to 44 zettabytes between 2013 and 2020. By 2025, IDC predicts there will be 163 zettabytes of data.

**Characteristics of Big Data**

Big data was originally associated with three key concepts: volume, variety, and velocity. Other concepts later attributed with big data are veracity and value.

**Volume:** The quantity of generated and stored data. The size of the data determines the value and potential insight, and whether it can be considered big data or not.

**Variety:** The type and nature of the data. This helps people who analyze it to effectively use the resulting insight. Big data draws from text, images, audio, video; plus it completes missing pieces through data fusion.

**Velocity:** In this context, the speed at which the data is generated and processed to meet the demands and challenges that lie in the path of growth and development. Big data is often available in real-time. Compared to small data, big data are produced more continually. Two kinds of velocity related to big data are the frequency of generation and the frequency of handling, recording, and publishing.

**Veracity:** It is the extended definition for big data, which refers to the data quality and the data value. The data quality of captured data can vary greatly, affecting the accurate analysis.

**1.2 DATA ANALYTICS**

Analytics is the discovery and communication of meaningful patterns in data. Especially, valuable in areas rich with recorded information, analytics relies on the simultaneous application of statistics, computer programming and operation research to qualify performance. Analytics often favors data visualization to communicate insight. Firms may commonly apply analytics to business data, to describe, predict and improve business performance. Especially, areas within include predictive analytics, enterprise decision management etc. Since analytics can require extensive computation (because of big data), the algorithms and software used to analytics harness the most current methods in computer science.

**Types of Data Analytics**

* Predictive (forecasting)
* Descriptive (business intelligence and data mining)
* Prescriptive (optimization and simulation)
* Diagnostic analytic

**Predictive Analytics**

Predictive analytics turn the data into valuable, actionable information. Predictive analytics uses data to determine the probable outcome of an event or a likelihood of a situation occurring. Predictive analytics holds a variety of statistical technique from modeling, machine learning, data mining and game theory that analyze current and historical facts to make prediction about future event. Predictive analytics is the branch of the advanced analytics which is used to make predictions about unknown future events. It uses both new and historical data to forecast, activity, behavior, and trends

There are three basic cornerstones of predictive analytics-

* Predictive modeling
* Decision Analysis and optimization
* Transaction profiling

**Descriptive Analytics**

Descriptive analytics looks at data and analyze past event for insight as how to approach future events. It looks at the past performance and understands the performance by mining historical data to understand the cause of success or failure in the past. Almost all the management reporting such as sales, marketing, operations, and finance uses this type of analysis. Descriptive model quantifies relationship in data in a way that is often used to classify customers or prospect into groups. Unlike predictive model that focuses on predicting the behavior of single customer, Descriptive analytics identify many different relationships between customer and product.

**Prescriptive Analytics**

Prescriptive Analytics automatically synthesize big data, mathematical science, business rule, and machine learning to make prediction and then suggests decision option to take advantage of the prediction. Prescriptive analytics goes beyond predicting future outcomes by also suggesting action benefit from the predictions and showing the decision maker the implication of each decision option. Prescriptive Analytics not only anticipates what will happen and when happen but also why it will happen. Further, Prescriptive Analytics can suggest decision options on how to take advantage of a future opportunity or mitigate a future risk and illustrate the implication of each decision option. For example, Prescriptive Analytics can benefit healthcare strategic planning by using analytics to leverage operational and usage data combined with data of external factors such as economic data, population demography etc.

**Diagnostic Analytics**

In this analysis, generally the historical data is used over other data to answer any question or for the solution of any problem. It is used to find any dependency and pattern in the historical data of the particular problem. For example, companies go for this analysis because it gives a great insight for a problem, and they also keep detailed information about the disposal otherwise data collection may turn out individual for every problem and it will be very time-consuming.

**Sentiment Analysis**

Sentiment Analysis is the process of ‘computationally’ determining whether a piece of writing is positive, negative or neutral. It’s also known as opinion mining, deriving the opinion or attitude of a speaker.In other words it is the process of using text analytics to mine various sources of data for opinions. Often, sentiment analysis is done on the data that is collected from the Internet and from various social media platforms. Politicians and governments often use sentiment analysis to understand how the people feel about themselves and their policies.

**Why sentiment analysis?**

**Business**: In marketing field companies use it to develop their strategies, to understand customers’ feelings towards products or brand, how people respond to their campaigns or product launches and why consumers don’t buy some products.

**Politics**: In political field, it is used to keep track of political view, to detect consistency and inconsistency between statements and actions at the government level. It can be used to predict election results as well!

**Public Actions**: Sentiment analysis also is used to monitor and analyze social phenomena, for the spotting of potentially dangerous situations.

**1.3 TWITTER DATA**

Twitter, one of the most popular social networking services in the world, is a micro blogging platform, which allows registered users to post, and send to other registered users, short messages. These short messages, limited to 140characters, are called Tweets. A tweet is a Twitter message displayed on a user's profile page, which is publicly visible by default, and shared with all his or her "followers." It can be described as a status update or post published by a Twitter user. Tweets are limited to 140 characters, including spaces, and may include URLs and hash tags.

**Twitter Sentiment Analysis**

Sentiment analysis involves language processing, text classification, and computational linguistics to extract emotional information from the source data. It is broadly employed to review the social media used in various fields such as marketing and customer service. Typically the intention of sentiment analysis is to estimate the mood of the user concerning the target object, and the basic task is to determine the polarity of given text. The approaches employed for sentiment analysis is roughly categorized into two types; machine learning-based and lexicon-based.

**1.4 MACHINE LEARNING TECHNIQUES**

Machine learning is a data analytics technique that teaches computers to do what comes naturally to humans and animals: learn from experience. Machine learning algorithms use computational methods to “learn” information directly from data without relying on a predetermined equation as a model. The algorithms adaptively improve their performance as the number of samples available for learning increases. With the rise in [big data](https://www.mathworks.com/solutions/big-data-matlab.html), machine learning has become a key technique for solving problems in areas, such as:

**Computational finance**, for [credit scoring](https://www.mathworks.com/discovery/credit-scoring-model.html) and [algorithmic trading](https://www.mathworks.com/discovery/algorithmic-trading.html).

**Energy production**, for price and [load forecasting](https://www.mathworks.com/discovery/load-forecasting.html).

**Automotive, aerospace, and manufacturing**, for [predictive maintenance](https://www.mathworks.com/discovery/predictive-maintenance.html).

**Natural language processing**, for voice recognition applications.

MACHINE LEARNING

SUPERVISED LEARNING

UNSUPERVISED LEARNING

CLUSTERING

CLASSIFICATION

REGRESSION

**Figure 1.1:**Types of Machine Learning

The above Figure 1.1 describes the types of Machine Learning Techniques. Unsupervised Learning uses Clustering whereas Supervised Learning technique uses Classification and Regression.

**1.4.1 Unsupervised Learning**

Unsupervised learning finds hidden patterns or intrinsic structures in data. It is used to draw inferences from datasets consisting of input data without labeled responses.

**Clustering**

Clustering is the most common unsupervised learning technique. It is used for exploratory data analysis to find hidden patterns or groupings in data. Applications for cluster analysis include gene sequence analysis, market research, and object recognition. Common algorithms for performing clustering include k-means and k-medoids, hierarchical clustering, Gaussian mixture models, hidden Markov models, self-organizing maps, fuzzy c-means clustering, and subtractive clustering.

**1.4.2 Supervised Learning**

Supervised learning builds a model that makes predictions based on evidence in the presence of uncertainty. A supervised learning algorithm takes a known set of input data and known responses to the data (output) and trains a model to generate reasonable predictions for the response to new data. Supervised learning uses classification and regression techniques to develop predictive models.

**1.5 CLASSIFICATION**

Classification techniques in data mining are capable of processing a large amount of data. It can be used to predict categorical class labels and classifies data based on training set and class labels and it can be used for classifying newly available data. The term could cover any context in which some decision or forecast is made on the basis of presently available information. Classification procedure is recognized method for repeatedly making such decisions in new situations. Classification techniques predict discrete responses—for example, whether an email is genuine or spam, or whether a tumor is cancerous or benign. Classification models classify input data into categories. Typical applications include medical imaging, speech recognition, and credit scoring. Common algorithms for performing classification include support vector machine (SVM), boosted and bagged decision trees, *k*-nearest neighbor, Naïve Bayes, discriminant analysis, logistic regression, and neural networks.

**Naive Bayes Classifier**

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of [feature](https://en.wikipedia.org/wiki/Feature_vector) values, where the class labels are drawn from some finite set. There is not a single [algorithm](https://en.wikipedia.org/wiki/Algorithm) for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is [independent](https://en.wikipedia.org/wiki/Independence_(probability_theory)) of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible [correlations](https://en.wikipedia.org/wiki/Correlation_and_dependence) between the color, roundness, and diameter features.

For some types of probability models, naive Bayes classifiers can be trained very efficiently in a [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning) setting. In many practical applications, parameter estimation for naive Bayes models uses the method of [maximum likelihood](https://en.wikipedia.org/wiki/Maximum_likelihood); in other words, one can work with the naive Bayes model without accepting [Bayesian probability](https://en.wikipedia.org/wiki/Bayesian_probability) or using any Bayesian methods.

**Support Vector Machine**

In [machine learning](https://en.wikipedia.org/wiki/Machine_learning), support-vector machines (SVMs, also support-vector networks) are [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning) models with associated learning [algorithms](https://en.wikipedia.org/wiki/Algorithm) that analyze data used for [classification](https://en.wikipedia.org/wiki/Statistical_classification) and [regression analysis](https://en.wikipedia.org/wiki/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](https://en.wikipedia.org/wiki/Probabilistic_classification) [binary](https://en.wikipedia.org/wiki/Binary_classifier) [linear classifier](https://en.wikipedia.org/wiki/Linear_classifier) (although methods such as [Platt scaling](https://en.wikipedia.org/wiki/Platt_scaling) exist to use SVM in a probabilistic classification setting). 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](https://en.wikipedia.org/wiki/Linear_classifier), SVMs can efficiently perform a non-linear classification using what is called the [kernel trick](https://en.wikipedia.org/wiki/Kernel_trick), implicitly mapping their inputs into high-dimensional feature spaces.

**Regression**

Regression techniques predict continuous responses—for example, changes in temperature or fluctuations in power demand. Typical applications include electricity load forecasting and algorithmic trading. Common regression algorithms includes linear model, nonlinear model, regularization, stepwise regression and [bagged](https://www.mathworks.com/help/stats/classification-ensembles.html) [decision trees](https://www.mathworks.com/help/stats/classification-trees.html), [neural networks](https://www.mathworks.com/help/nnet/function-approximation-and-nonlinear-regression.html), and [adaptive neuro-fuzzy learning](https://www.mathworks.com/help/fuzzy/anfis.html).

**CHAPTER 2**

**LITERATURE SURVEY**

In this section the literature survey has been carried out. The main focus is on text classification techniques on sentiment analysis. The literature survey gives a clear idea for proposed system.

Bahassine S et al [1] imparted “An improved Chi-square feature selection for Arabic text classification using decision tree”. They investigated a new feature selection (referred to, hereafter, as Imp CHI), when using light stemming. Imp CHI is an improvement of chi-square - one of the most effective feature selection methods to date. Evaluation used a corpus that consists of 250 Arabic documents independently classified into five classes: art and culture, economics, politics, society, and sport. Their experiment results showed that Arabic text classification using Imp CHI as feature selection outperforms using chi-square in terms of recall-measures.

Gu B et al [2] presented “A robust regularization path algorithm for ν-support vector classification”. The v-support vector classification has the advantage of using a regularization parameter v to control the number of support vectors and margin errors. They found a regularization path algorithm for v-support vector classification (v-SvcPath) suffers exceptions and singularities in some special cases. The work includes new equivalent dual formulation for v-SVC and, then proposed a robust v-SvcPath, based on lower upper decomposition with partial pivoting. Theoretical analysis and experimental results verify that their proposed robust regularization path algorithm can avoid the exceptions completely, handle the singularities in the key matrix, and fit the entire solution path in a finite number of steps. Experimental results also showed that the proposed algorithm fits the entire solution path with fewer steps and less running time than original one does.

Jiadong Z et al [3] represented a “Feature extraction for robust physical activity recognition”. The purpose of their work is to present the development of a Human Activity Recognition (HAR) system that uses a network of nine inertial measurement unit situated in different body parts. Every unit provides 3D (3-Dimension) acceleration, 3D angular velocity, 3D magnetic field orientation, and 4D quaternion’s. The system identifies 33 different physical activities (walking, running, cycling, lateral elevation of arms, etc.). The system is composed of two main modules: a feature extractor for obtaining the most relevant characteristics from the inertial signals every second, and a machine learning algorithm for classifying between the different activities. They focus on the feature extractor module, evaluating several types of features and proposing different normalization approaches. And also analyzed the performance of every sensor included in the inertial measurement units.

Krouska A et al [4] intimated “The effect of preprocessing techniques on twitter sentiment analysis”. They explained the necessary information to get preprocess the reviews in order to find sentiment and make analysis whether it is positive or negative. Extended comparison of sentiment polarity classification methods for Twitter text and the role of text preprocessing in sentiment analysis were discussed in depth. The set of tests includes possible combinations of methods and report on their efficiency, conducting experiments using manually annotated Twitter datasets. Finally, they proved that the feature selection and representation can affect the classification performance positively.

Krouska A et al [5] presented a “Comparative evaluation of algorithms for sentiment analysis over social networking services”. Twitter represents one of the largest and most dynamic datasets for data mining and sentiment analysis. Therefore, Twitter Sentiment Analysis constitutes a prominent and an active research area with significant applications in industry and academia. They provided a guideline for the decision of optimal algorithms for sentiment analysis services. The context contains five well-known learning-based classifiers (Naive Bayes, Support Vector Machine, k-Nearest Neighbor, Logistic Regression and C4.5) and a lexicon-based approach (SentiStrength) that have been evaluated based on confusion matrices, using three different datasets and two test models (percentage split and cross validation). The results demonstrated the superiority of Naive Bayes and Support Vector Machine regardless of datasets and test methods.

Lizhen L et al [6] imparted “A feature based method for Sentiment Analysis for Chinese Product Reviews”. They proposed a feature-based vector model and a novel weighting algorithm for sentiment analysis of Chinese product reviews. Their work considered modifying relationships between words and contains rich sentiment strength descriptions which are represented by adverbs of degree and punctuations. A novel feature weighting algorithm was proposed for supervised sentiment classification based on rich sentiment strength related information. The experimental results demonstrated the effectiveness of the proposed method compared with a state of the art method using term level weighting algorithms.

Paul S et al [7] represented “A simultaneous feature selection and weighting–an evolutionary multi-objective optimization approach”. They introduced a new feature selection and weighting method aided with the decomposition based evolutionary multi-objective algorithm called MOEA/D. The feature vectors are selected and weighted or scaled simultaneously to project the data points to such a hyper space, where the distance between data points of non-identical classes is increased, thus, making them easier to classify. The inter-class and intra-class distances are simultaneously optimized by using MOEA/D to obtain the optimal features and the scaling factor associated with them. Finally, k-NN (k-Nearest Neighbor) is used to classify the data points having the reduced and weighted feature set. The proposed algorithm was tested with several practical datasets from the well-known data repositories like UCI and LIBSVM. The results are compared with those obtained with the state-of-the-art algorithms to demonstrate the superiority of the proposed algorithm.

Salton G et al [8] presented “A Term-weighting approaches in automatic text retrieval”. The experimental evidence accumulated over the past 20 years indicates that text indexing systems based on the assignment of appropriately weighted single terms produce retrieval results that are superior to those obtainable with other more elaborate text representations. Those results depend crucially on the choice of effective term weighting systems. They summarized the insights gained in automatic term weighting, and provide baseline single-term-indexing models with which other more elaborate content analysis procedures can be compared.

Sebastiani F [9] reported “A machine learning in automated text categorization”. The automated categorization (or classification) of texts into predefined categories has witnessed a booming interest in the last 10 years, due to the increased availability of documents in digital form and the ensuing need to organize them. The dominant approach to their problem is based on machine learning techniques: a general inductive process automatically builds a classifier by learning, from a set of preclassified documents, the characteristics of the categories. The advantages of the approach over the knowledge engineering approach (consisting in the manual definition of a classifier by domain experts) are a very good effectiveness, considerable savings in terms of expert labor power, and straightforward portability to different domains. The survey discussed the main approaches to text categorization that fall within the machine learning paradigm.

Suresh Y [10] intimated “Quality of a software product being designed, has a critical role in software process management”. Detection and prediction of faults in software with huge lines of code is a very tedious task. So it is very essential, as to reduce the maintenance cost and in turn increase the software reliability. Many object-oriented metrics have found to be suitable for software fault prediction. Using data mining techniques, design of prediction and classification models can be incorporated to give insight of the systems quality to the developing team to effectively tackle the quality problems. Chidamber and Kemerer Metric suite, along with classifiers which have immense classification capacity have been used in predicting software fault classification accuracy. Finally, they concluded that Logistic classifier is able to obtain better fault classification accuracy when compared to Naive Bayes approach.

Wen Z et al [11] presented “A comparative study of TF\* IDF, LSI and multi-words for text classification”. They has comparatively studied TF\*IDF, LSI and multi-word for text representation and used a Chinese and an English document collection to respectively evaluate the three methods in information retrieval and text categorization. Experimental results have demonstrated that in text categorization, LSI has better performance than other methods in both document collections. Also, LSI has produced the best performance in retrieving English documents. Their outcome has shown that LSI has both favorable semantic and statistical quality and is different with the claim that LSI cannot produce discriminative power for indexing.

Wilson T et al [12] imparted a “Recognizing contextual polarity in phrase-level sentiment analysis”. They introduced a new approach to phrase-level sentiment analysis that first determines whether an expression is neutral or polar and then disambiguates the polarity of the polar expressions. With that approach, the system is able to automatically identify the polarity for a large subset of sentiment expressions, achieving results that are significantly better than baseline.

Yu Y et al [13] presented” A big data analysis of sentiments in US sports fans”. They used sentiment analysis to examine U.S. soccer fans’ emotional responses in their tweets, particularly, the emotional changes after goals (either own or the opponent’s). They found that during the matches that the U.S. team played, fear and anger were the most common negative emotions and in general, increased when the opponent team scored and decreased when the U.S. team scored. Anticipation and joy were also generally consistent with the goal results and the associated circumstances during the games. Furthermore, they found that during the matches between other teams, U.S. tweets showed more joy and anticipation than negative emotions (e.g., anger and fear) and that the patterns in response to goal or loss were unclear. They revealed that sports fans use Twitter for emotional purposes and that the big data approach to analyze sports fans’ sentiment showed results generally consistent with the predictions of the disposition theory when the fan ship was clear and showed good predictive validity.

Yuhui Z et al [14] presented a “Student’s t-hidden Markov model for unsupervised learning using localized feature selection”. A novel SHMM is proposed by combining the measure of localized feature saliency (LFS) with SMM and utilizing two student's t-distributions as subcomponents to respectively describe the distributions of useful features and non-salient “features,” with the purpose of accurately modeling the hidden state observation emission distributions of SHMM. Moreover, they exploit the variational Bayesian learning technique to simultaneously estimate the LFS, the number of components and other parameters of the herein proposed SHMM. Experimental results on both synthetic and real data sets demonstrate the improved robustness, effectiveness, and accuracy of our model.

Zou H et al [15] represented a “Sentiment classification using machine learning techniques with syntax features”. Sentiment classification has adopted machine learning techniques to improve its precision and efficiency. However, the features are always produced by basic words-bag methods without much consideration for words' syntactic properties, which could play an important role in the judgment of sentiment meanings. To remedy that, they firstly generated syntax trees of the sentences, with the analysis of syntactic features of the sentences. Then they introduced multiple sentiment features into the basic words-bag features. Such features were trained on movie reviews as data, with machine learning methods (Naive Bayes and support vector machines). The features and factors introduced by syntax tree were examined to generate a more accurate solution for sentiment classification.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

* The existing system provides a novel feature weighting approach for Sentiment Analysis of twitter data.
* The existing system uses Bayes based classifier and has classified the Twitter data.
* Extensive experiments were conducted on Sentiment 140, and four representative feature weighting schemes were also tested to demonstrate the performance.
* The system also provides us informative data for classification by choosing effective feature selection strategy recognizing sentiment sentence.
* Each word are grouped into target cluster considering POS property of the word.
* This scheme consistently outperforms others in terms of accuracy, precision, recall, and F1-measure.

**3.1.1 Limitations**

* Performance of Sentiment analysis is not efficient.
* Polarity of words is not much accurate.

**3.2 PROPOSED SYSTEM**

The scheme of POS tagging is useful in classifying the words into emotional and normal classes. POS tagger adds tags to the words in a sentence such as adverb, adjective, noun and verb. This tags helps to classify the words into emotional and normal classes. Adjective, Adverb and verb are considered as emotional words. Then feature clustering and weighting is performed based on predefined tags. Feature weighting is done by modified CHI Square method. This method computes independency between features. It overcomes the overemphasizing of the words. This system focuses mainly to show better accuracy in terms of polarity of words. This accuracy can be achieved by testing the existing scheme against another classifier that is Support Vector Machine (SVM). SVM uses a hyperplane which separates two classes, so that classification performed accurately.

**CHAPTER 4**

**SYSTEM SPECIFICATION**

**4.1 SOFTWARE REQUIREMENTS**

Operating System : Windows Pro

Front End : Anaconda Navigator IDE

Back End : My SQL

Language : Python

**4.2 HARDWARE REQUIREMENTS**

Processor : Intel Core i3

Hard Disk : 512 GB

RAM : 4 GB

**4.3 ABOUT THE SOFTWARE**

**Anaconda Navigator**

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda® distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, macOS and Linux. Anaconda® is a package manager, an environment manager, a Python/R data science distribution, and a collection of over 1500+ open source packages. Anaconda is free and easy to install, and it offers free community support. Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window.

**Conda**

Conda is an open source package management system and environment management system that runs on Windows, macOS and Linux. Conda quickly installs runs and updates packages and their dependencies. Conda easily creates, saves, loads and switches between environments on your local computer. It was created for Python programs, but it can package and distribute software for any language.

Conda can be combined with continuous integration systems such as Travis CI and AppVeyor to provide frequent, automated testing of your code. The conda package and environment manager is included in all versions of Anaconda and Miniconda, Anaconda Repository. Conda is also included in Anaconda Enterprise , which provides on-site enterprise package and environment management for Python, R, Node.j.

**Applications Available in Navigator**

* + - JupyterLab
    - Jupyter Notebook
    - QTConsole
    - Spyder
    - VSCode
    - GlueViz

**Jupyter Notebook**

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

**Python**

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently. There are two major Python versions- **Python 2 and Python 3**. Both are quite different. Some of the advantages of using python:

* Emphasis on code readability, shorter codes, ease of writing
* Programmers can express logical concepts in fewer linesof code in comparison to languages such as C++ or Java.
* Python supports multipl**e** programming paradigms, like object-oriented, imperative and functional programming or procedural.
* There exist inbuilt functions for almost all of the frequently used concepts.

**Features**

* Interpreted
* Platform Independent
* Free and open source, Redistributable
* Embeddable
* Robust
* Rich Library support

**Data Analysis in Python**

Python is a great language for doing data analysis, primarily because of the fantastic ecosystem of data-centric Python packages. Pandas is one of those packages, and makes importing and analyzing data much easier.

**NLTK**

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources. Such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

NLTK is suitable for linguists, engineers, students, educators, researchers, and industry users alike. NLTK is available for Windows, Mac OS X, and Linux. Best of all, NLTK is a free, open source, community-driven project.

**CHAPTER 5**

**SYSTEM DESIGN**

**5.1 SYSTEM ARCHITECTURE**

Polarity of data

Tweets

Classifier

(SVM)

Feature Weighting

(CHI Square and WF)

Text

Text

Normalization

Feature Clustering

Test Data

Sentence Recognition

(POS Tagger)

Training Dataset

**Figure 5.1:** Architecture of the proposed system

The above Figure 5.1 shows the Architectural diagram of the proposed system. Here the text is taken from the tweets and normalized. POS tagging is added to each words in the text and Feature clustering and weighting is performed. After that the training and test data is tested using classifiers to give the polarity of the data.

* 1. **MODULE DESCRIPTION**

The Proposed System consists of the following modules:

* Dataset Collection
* Data Preprocessing
* Feature Clustering
* Feature Weighting

**5.2.1 Dataset Collection**

Data is a piece of information that should be collected carefully so that the collected information is useful. Data collection is an important step while doing experiments or researches. Data collection is the process of gathering or processing information that is used for obtaining outcomes in experiments. Social Data’s are those that can be collected from the social media platform. Twitter is the online micro blogging service where users such as individuals and organizations share their views of their interest. These views and updates contains 140 characters as length of tweet. On Twitter, results include numbers of impressions or re-tweets.The collection of dataset is the major part in the proposed system. Here we collect data’s in the form of tweets. Data’s are collected from twitter API. The data’s are preprocessed and refined into CSV format. The preprocessing involves Removal of stop words, rare words, normal words which are not emotional. Further the preprocessed data is loaded into Anaconda Navigator.

**5.2.2 Data Preprocessing**

The following Figure 5.2 shows the steps of preprocessing. This preprocessing involves transforming raw data into an understandable format.Real world data often contains noise, incomplete, inconsistent data, lacking in certain patterns. These issues are resolved by Data Preprocessing.

Positive and Negative Status Update

Text

Normalization

Training Dataset

**Figure 5.2:** Preprocessing steps

Preprocessing steps involves Data Cleaning, Data Integration and Data Transformation. Text Data is preprocessed via following steps:

* Punctuation Removal
* Stop Words Removal
* Frequent Words Removal
* Rare Words Removal
* Tokenization (Separation of words in a Corpora)
* Stemming (Removal of suffices)
* Lemmatization (Converting into root words after stemming)

**5.2.3 Feature Clustering**

It is the most important tasks related to classification. It includes the removal of irrelevant words or terms that do not express any sentiment. Unigram (n=1) and term frequency and inverse document frequency (TF-IDF) is used for feature extraction. The unigram represents individual and distinct words. The TF-IDF assigns a score to each word. The term- frequency is computed by counting the number of times a given word or term appeared in given document and inverse document frequency is computed by dividing the total number of documents by number of documents that has a given term.

The sentences of twitter in the training data are classified into positive and negative sentences. The emotional words in a sentence are important in predicting the sentiment of a sentence and hence removal of sentences having few emotional words can improve the accuracy of Sentiment Analysis. Here the features are selected based on word frequency that follows POS tagging of the words. The POS tagging is a method of splitting the sentences into words and attaching a proper tag such as noun, verb, adjective, adverb to each word based on POS tagging rules. Here Adverb, Adjective and Verb are regarded as emotional features in deciding the sentiment and the sentence containing less than two types of emotional features is considered as unrelevant to sentiment analysis, thus removed from training dataset. For efficient feature clustering, the unigram feature extractor is utilized to retrieve the features which treat each unique word in the training dataset as a unit representing separate features. Here the feature set is classified into two classes Emotional and Normal. As mentioned above the words which are adjective, adverb and verb is regarded as emotional class, and remaining are considered as normal class.

**5.2.4 Feature Weighting**

Word weightingis performed based on the importance of the word in the training dataset. This scheme considers the dependency of the clustered feature set with the class, which is measured by the Chi square method. A modified Chi Square method is employed since the conventional Chi Square method suffers from the shortcoming of overemphasizing the role of the words of low frequency. Therefore, WF is proposed to serve as the input to the Chi Square method to overcome such weakness. A novel composite Feature Weighting technique is proposed, which considers the dependency derived using the modified Chi Square method and discriminability of the clustered feature set.

At the same time the importance of words of strong discriminability is emphasized in the weighting process so that they can take more significant role in the sentiment analysis. The term frequency and inverse document frequency (TF–IDF) is a commonly adopted feature weighting scheme owing to its efficiency and robustness. It assumes that importance of a word is highly dependent on its frequency of occurrences in the document and the ratio of the total number of documents to the number of documents containing the word.

**Testing**

Sentiment analysis is a subfield of NLP concerned with the determination of opinion and subjectivity in a text, which has many applications. Support Vector Machines are based on the concept of decision planes that define decision boundaries. A decision plane is one that separates between a set of objects having different class memberships. Support Vector Machine (SVM) is primarily a classier method that performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels. SVM is generally used for text categorization. It can achieve good performance in high-dimensional feature space. An SVM algorithm represents the examples as points in space, mapped so that the examples of the different categories are separated by a clear margin as wide as possible. It gives best results than Naive Bayes algorithm and sentiment classification tools. The basic idea is to find the hyperplane which is represented as the vector w which separates document vector in one class from the vectors in other class. SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables. For categorical variables a dummy variable is created with case values as either 0 or 1. Thus, a categorical dependent variable consisting of three levels, say (A, B, C), is represented by a set of three dummy variables:

A: {1 0 0}, B: {0 1 0}, C: {0 0 1}

To construct an optimal hyperplane, SVM employs an iterative training algorithm, which is used to minimize an error function. According to the form of the error function, SVM models can be classified into four distinct groups:

* + - * Classification SVM type 1
      * Classification SVM type 2
      * Regression SVM type 1
      * Regression SVM type 2

**Mathematical Model of SVM**

* Let S be the system that describes the tweet extraction, Preprocessing, Sentiment labeling, Sentiment Variation Tracking and Reason candidate for Sentiments.
* S= {Tw, Pt, Sl, Vt, Rs}
* Tw =Tweets extracted from Twitter.
* Pt =Preprocessing of Tweets (Slang word translation, Non-English word removal, PoS tagging, URL and Stop word removal).
* Sl=Sentiment Labeling using SentiStrength and TwitterSentiment sentiment analysis tools (SVM to give more efficient and accurate results).
* Sl={Pv, Nv, Ne}
* Pv={P1,P2,…,Pn}= Positive Class
* Nv={ N1,N2,…,Nn }= Negative Class
* Ne={ Ne1,Ne2,…,Nen }= Neutral Class
* Vt =Sentiment Variation Tracking
* Rs=Reason Candidate for Sentiment Variations using RCB-LDA model .

|  |
| --- |
|  |

**5.3 EXPERIMENTAL RESULTS**

The workload obtained is used to analyze the accuracy of the proposed scheme for twitter sentiment analysis. The simulator consists of three parts; preprocessor, POS tagger, and SVM classifier. The preprocessor classifies the data of the training data set, and converts them to the customized format accessible by the POS tagger. The SVM classifier is used to classify the tested document and predict the sentiment of the target sentences. The workload used in the simulation cosist of 10,000 lines of tweet data. The data set consist of about 5000 negative class and the remaining are positive class. The proposed scheme consistently outperforms the other schemes regardless of the size of training. Intuitively, the accuracy of sentiment analysis increases as the volume of training data grows. It is because the larger the training data, the more evidences could be provided for sentiment judgment.

Also notice that the accuracy of the proposed scheme gradually increases with the growth of the size of training data set. The feature weighting scheme proposed here showed a better performance due to the reason that the scheme gives importance to the words in the training dataset. This enhances the feature of important words of high discriminability, which in turn produces higher accuracy than other schemes. The proposed scheme substantially displays higher accuracy than the existing system. Note that the existing method significantly outperforms other schemes for the three well-known benchmarks. The existing scheme produced best performance in terms of precision, recall, and F1-measures scheme offers good accuracy when the size of training data is large. Our proposed scheme performed well and resulted in better accuracy as compared to the existing scheme.

**CHAPTER 6**

**CONCLUSION AND FUTURE WORKS**

**6.1 CONCLUSION**

Twitter Sentiment Analysis has become a promising technique for industry and academia. The proposed system uses Support Vector Machine for sentiment analysis of twitter data to increase the accuracy. Features clustered based on POS tagging has improved the efficiency of sentiment analysis The existing system used Bayes based classifier which assumes that there exists a independency between features. Whereas SVM uses a hyperplane which separates training data into two classes which results in better accuracy in terms of polarity.

**6.2 FUTURE WORKS**

Classification algorithm relies more on the sentiment analysis of twitter data to yield a prediction about research. Future works should consider additional parameters and updates in twitter API to give a better accuracy. They should also consider the issues related to classification. And also this scheme will be tested using various classifiers such as Decision tree and Neural Networks.

**APPENDIX-I**

**SAMPLE SOURCE CODE**

**Reading csv file**

Import pandas as pd

train = pd.read\_csv('tweet\_train.csv')

**Counting of words**

train['word\_count'] = train['TWEET'].apply(lambda x: len(str(x).split(" ")))

train[['TWEET','word\_count']].head

**Counting of characters**

train['char\_count'] = train['TWEET'].str.len()

train[['TWEET','char\_count']].head

**Average word length**

def avg\_word(sentence):

words = sentence.split()

return (sum(len(word) for word in words)/len(words))

train['avg\_word'] = train['TWEET'].apply(lambda x: avg\_word(x))

train[['TWEET','avg\_word']].head

**Stop Words Removal**

import nltk

from nltk.corpus import stopwords

stop = stopwords.words('english')

train['stopwords'] = train['TWEET'].apply(lambda x: len([x for x in x.split() `if x in stop]))

train[['TWEET','stopwords']].head

nltk.download('punkt')

from nltk.corpus import stopwords

stop = stopwords.words('english')

train['TWEET'] = train['TWEET'].apply(lambda x: " ".join(x for x in x.split() if x not in stop))

train['TWEET'].head

**Frequently occurring words removal**

freq = pd.Series(' '.join(train['TWEET']).split()).value\_counts()[:10]

freq

freq = list(freq.index)

train['TWEET'] = train['TWEET'].apply(lambda x: " ".join(x for x in x.split() if x not in freq))

train['TWEET'].head

**Rare Words Removal**

freq = pd.Series(' '.join(train['TWEET']).split()).value\_counts()[-10:]

freq

freq = list(freq.index)

train['TWEET'] = train['TWEET'].apply(lambda x: " ".join(x for x in x.split() if x not in freq))

train['TWEET'].head()

**Stemming**

TextBlob(train['TWEET'][101]).words

from nltk.stem import PorterStemmer

st = PorterStemmer()

train['TWEET'][:100].apply(lambda x: " ".join([st.stem(word) for word in x.split()]))

**Lemmatization**

from textblob import Word

train['TWEET']=train['TWEET'].apply(lambdax:"".join([Word(word).lemmatize() for word in x.split()]))

train['TWEET'].head

TextBlob(train['TWEET'][0]).ngrams(1)

train['TWEET'][:5].apply(lambda x: TextBlob(x).sentiment)

train['sentiment']= train['TWEET'].apply(lambda x: TextBlob(x).sentiment[0 ] )

train[['TWEET','sentiment']].head

**POS Tagging**

text=train['TWEET'].str.split().map(pos\_tag)

text.head()

**Splitting Datasets**

Train\_X, Test\_X, Train\_Y, Test\_Y = model\_selection.train\_test\_split(Corpus['text\_final'],Corpus['label'],test\_size=0.3)

Encoder = LabelEncoder()

Train\_Y = Encoder.fit\_transform(Train\_Y)

Test\_Y = Encoder.fit\_transform(Test\_Y)

**TF-IDF**

Tfidf\_vect = TfidfVectorizer(max\_features=1500)

Tfidf\_vect.fit(train['TWEET'])

Train\_X\_Tfidf = Tfidf\_vect.transform(Train\_X)

Test\_X\_Tfidf = Tfidf\_vect.transform(Test\_X)

print(Tfidf\_vect.vocabulary\_)

print(Train\_X\_Tfidf)

**Clssification Model**

SVM = svm.SVC(C=1.0, kernel='linear', degree=3, gamma='auto')

SVM.fit(Train\_X\_Tfidf,Train\_Y)

pred=SVM.predict(Test\_X\_Tfidf)

pred

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import accuracy\_score

from sklearn.metrics import classification\_report

actual = Test\_Y

predicted = pred

results = confusion\_matrix(actual, predicted)

print('Confusion Matrix :')

print(results)

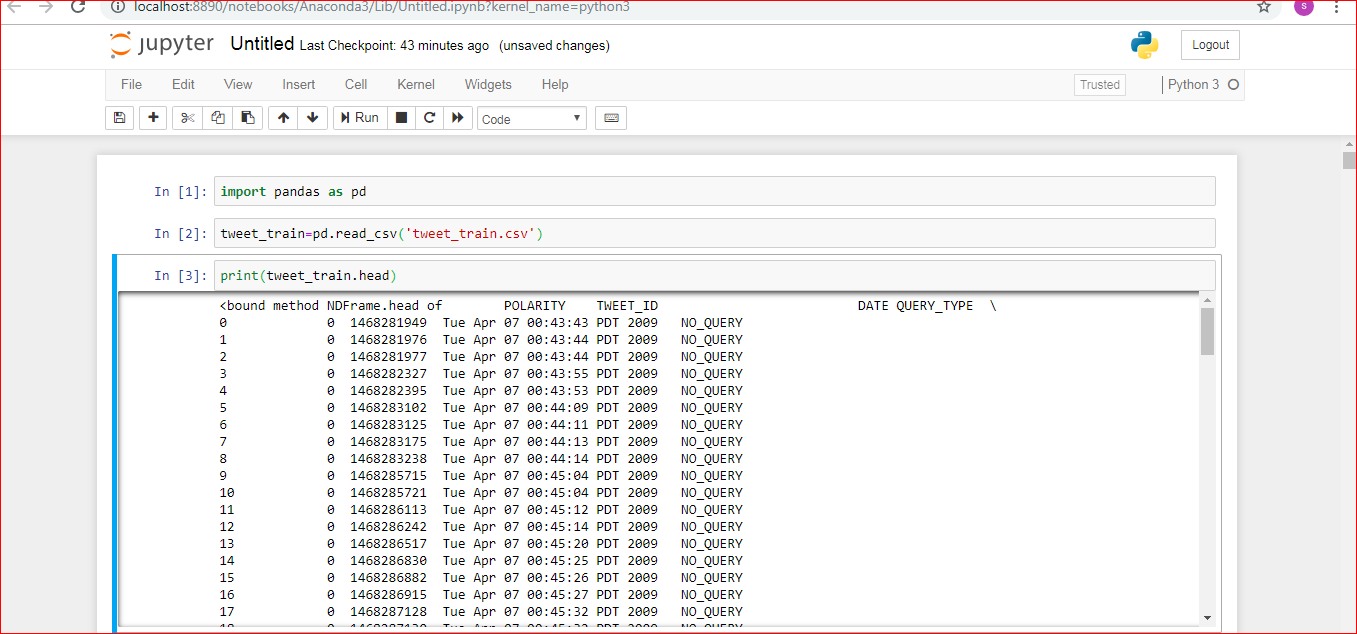
print('Accuracy Score :',accuracy\_score(actual, predicted))

print('Report : ')

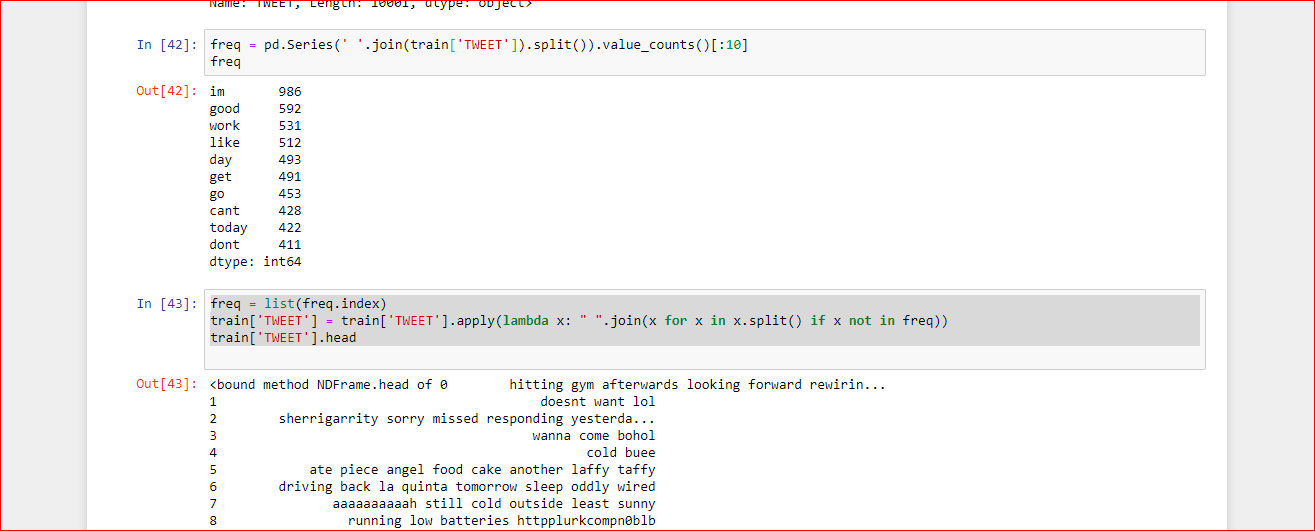
print(classification\_report(actual, predicted))

**APPENDIX-II**

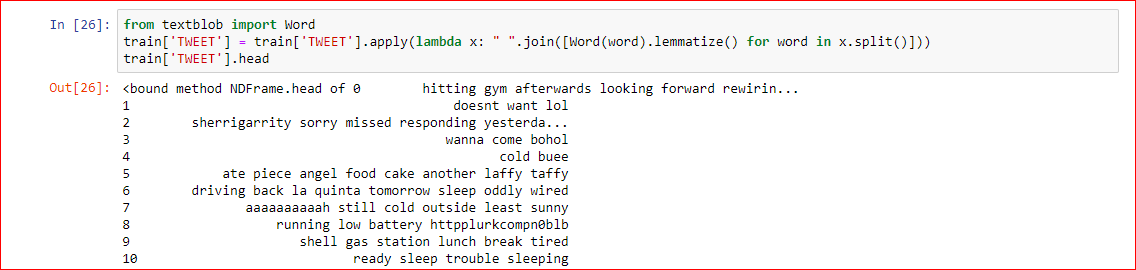
**SAMPLE SCREENSHOTS**

****

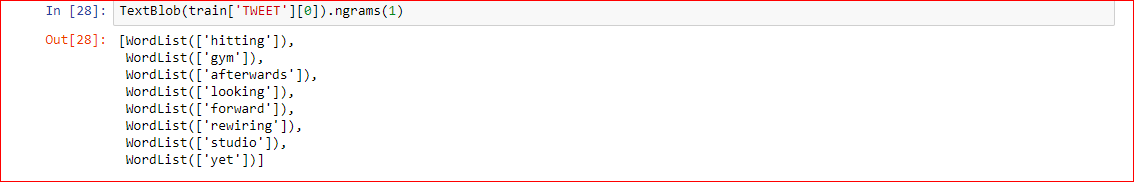
**S1 : Reading Dataset**

****

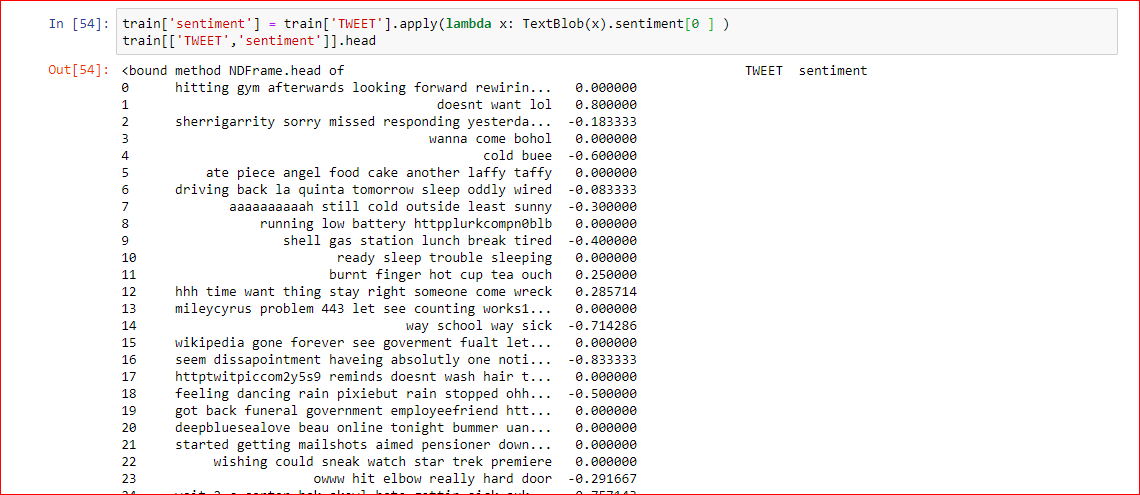
**S2 : Rare Words Removal**

****

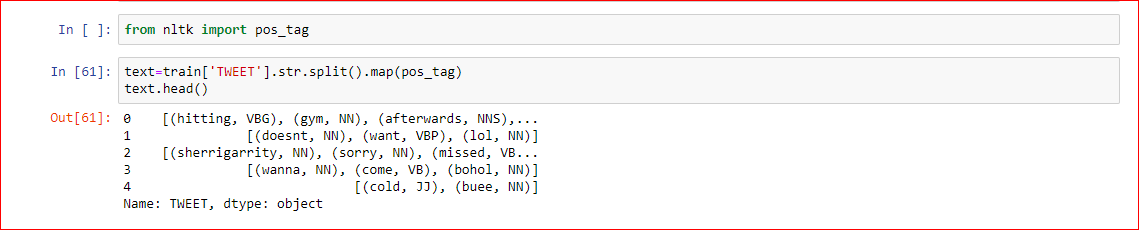
**S3 : Lemmatization of Words**

****

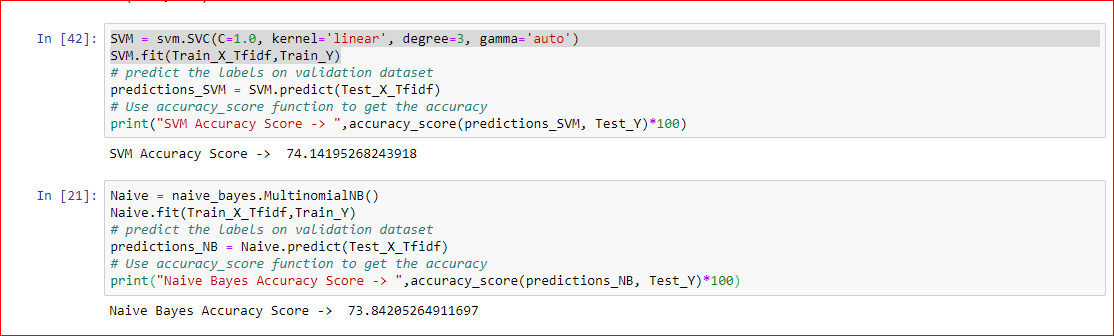
**S4 : Unigram Feature Extraction**

****

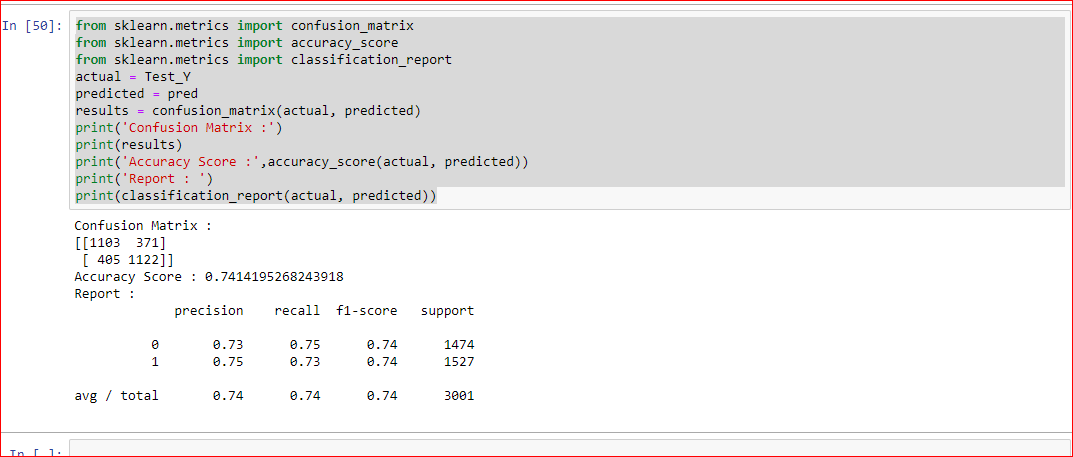
**S5: Sentiment Accuracy of Words**

****

**S6: POS Tagging**

****

**S7: Comparing SVM and Naive Bayes Classifiers**

****

**S8: Confusion Matrix**

**REFERENCES**

[1] Bahassine S, Madani A, Kissi M (2016) An improved Chi-sqaure feature selection for Arabic text classification usingdecision tree. Paper presented at the international conference on intelligent systems: theories and applications, 1–5Oct 2016.

[2] Gu B, Sun X, Sheng VS (2017) Structural minmax probability machine. IEEE Trans Neural Network Learning System 28:1646–1656.

[3] Jiadong Z, San-Segundo R, Pardo JM (2017) Feature extraction for robust physical activity recognition. Hum Comput Inf Sci 7:16.

[4] Krouska A, Troussas C, Virvou M (2016) The effect of preprocessing techniques on twitter sentiment analysis. Paper presented at the 7th international conference on information, intelligence, systems and applications, 1–5 December 2016.

[5] Krouska A, Troussas C, Virvou M (2016) The effect of preprocessing techniques on twitter sentiment analysis. Paper presented at the 7th international conference on information, intelligence, systems and applications, 1–5 July 2016.

[6] Lizhen L et al (2014) A novel feature-based method for sentiment analysis of Chinese product reviews. ChinaCommun11:154–164.

[7] Paul S, Das S (2015) simultaneous feature selection and weighting–an evolutionary multi-objective optimization approach. Pattern Recognit Lett 65:51–59.

[8] Salton G, Buckley C (1988) Term-weighting approaches in automatic text retrieval. Inf Pro Man 24:513–523.

[9] Sebastiani F (2002) machine learning in automated text categorization. Paper presented at the ACM computing surveys, 34:1–47 March 2002.

[10] Suresh Y (2016) Software quality assessment for open source software using logistic and Naive Bayes classifier. Paper presented at the International conference on computation system and information technology for sustainable solutions,267–272 Oct 2016.

[11] Wen Z, Yoshida T, Xinjing T (2011) a comparative study of TF\* IDF, LSI and multi-words for text classification. Expert System Application 38:2758–2765.

[12] Wilson T, Wiebe J, Hoffmann P (2005) Recognizing contextual polarity in phrase-level sentiment analysis. Paper presented at the conference on human language technology and empirical methods in natural language processing, 347–354 Oct 2015.

[13] Yu Y, Xiao W (2015) World Cup 2014 in the twitter World: a big data analysis of sentiments in US sports fans. Comput Hum Behav 48:392–400.

[14] Yuhui Z et al (2017) Student’s t-hidden Markov model for unsupervised learning using localized feature selection. IEEE Transaction Circuits System Video Technology.

[15] Zou H et al (2015) Sentiment classification using machine learning techniques with syntax features. Paper presented at the international conference on computational science and computational intelligence, 175–179 March 2015.