### MACHINE LEARNING:TEXT CLASSIFICATION

### USING PYTHON

## A PROJECT REPORT

## *submitted to*

**SRI VENKATESWARA UNIVERSITY COLLEGE OF ENGINEERING**

**In partial fulfilment of requirements for the award of the degree of**

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

***by***

**P. SANA MUBEEN (11506027)**

**R. SASI KUMAR REDDY (11506034)**

**M.PAVANI (11506041)**

***Under the esteemed guidance of***

**Dr. P. VENKATA SUBBA REDDY**

## Professor & Chairman

## Department of Computer Science & Engineering



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING SRI VENKATESWARA UNIVERSITY COLLEGE OF ENGINEERING, TIRUPATI-517502, 2018-19.**

**SRI VENKATESWARA UNIVERSITY COLLEGE OF ENGINEERING, TIRUPATI – 517502.**



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING CERTIFICATE**

This is to certify that the project entitled ― **TEXT CLASSIFICATION USING MACHINE LEARNING ALGORITHMS is** genuine and has been carried out under my supervision in the **Department of Computer Science and Engineering**, **Sri Venkateswara University College of Engineering.**

The work is comprehensive, complete and fit for evaluation carried out in partial fulfilment of the requirements for the award of **Bachelor of Technology** in **Computer Science and Engineering** during the academic year2018-19.

**P. SANA MUBEEN (11506027)**

**R. SASI KUMAR REDDY (11506034)**

**M. PAVANI (11506041)**

To the best of my knowledge matter embodied in the thesis has not been submitted to any other University/Institution for the award of any Degree or Diploma.

**GUIDE: HEAD OF THE DEPT.:**

**Dr. P VENKATA SUBBA REDDY Dr. M. HUMERA KHANAM**

Prof. Department of CSE, SVUCE

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**SRI VENKATESWARA UNIVERSITY COLLEGE OF ENGINEERING, TIRUPATI – 517502.**



Declaration

This is to certify that the project entitled ―**TEXT CLASSIFICATION USING MACHINE LEARNING ALGORITHM** is a bonafide work performed by us, the students mentioned below for the partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** from **Sri Venkateswara University College of Engineering.**

To the best of our knowledge, this project report has not been submitted to any other institution/ University.

**P. SANA MUBEEN (11506027)**

**R. SASI KUMAR REDDY (11506034)**

**M.PAVANI (11506041)**

# ACKNOWLEDGEMENTS

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of my project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

First and foremost, we would like to express our gratitude and indebtedness to our project guide **Dr. P. VENKATA SUBBA REDDY,** who took keen interest on our project work and guided us all along, till the completion of our project work by providing all the necessary information for developing a good system. We are sincerely thankful for his immense help, encouragement and belief in us, and for his inspiring guidance, constructive criticism and priceless suggestions throughout this project work.

We are deeply obliged to **Dr. M. HUMERA KHANAM,** Head of the Department, CSE, for providing valuable suggestions and critical reviews on the project, after project work and helping us in improving and providing the necessary administrative help and for providing required facilities in order to complete the project.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of CSE which helped us in successfully completing our project work.

We would like to express thanks to our parents for their continued support & encouragement through the ups & downs encountered on the path to completing this project.

Thanking you,

**P. SANA MUBEEN (11506027)**

**R. SASI KUMAR REDDY (11506034)**

**M. PAVANI (1150641)**

#### **ABSTRACT**

Automated text classification has been considered as a vital

method to manage and process a vast amount of documents in digital forms that are widespread and continuously increasing. In general, text classification plays important role in information extraction and summarization, text retrieval, and question answering. Sentiments are expressions of one’s words in a sentence. Hence understanding the meaning of text in the sentence is of utmost importance to people of various fields like customer reviews in companies, movie reviews in movies etc.

Supervised classification of text is done when you have defined the classification categories. It works on training and testing principle. It may involve huge text data to analyze and it becomes totally unviable for manually understanding the meaning of sentences. Classifier algorithms should be used to classify the various meaning of the sentences. By using predefined data to train our classifier, Naïve Bayes algorithms (Gaussian Naïve Bayes, Bernoulli Naïve Bayes & Multinomial Naïve Bayes Algorithm). This illustrates the text classification process using machine learning techniques.

##### **CONTENTS**

##### PAGE NO.

##### DECLARATION 01-04

##### ABSTRACT 05

##### CONTENTS 06

##### LIST OF FIGURES 07

##### 

CHAPTER 1 INTRODUCTION 09-17

* 1. PURPOSE 09
  2. INTENDED AUDIENCE 11
  3. PROJECT SCOPE 12
  4. OPERATING ENVIRONMENT 15
  5. SYSTEM AND DESIGN CONSTRAINTS 16

CHAPTER 2 SYSTEM FEATURES 18-42

2.1 SYSTEM DESIGN AND MODEL 18

CHAPTER 3 INTERFACE REQUIREMENTS 43-50

3.1 USER INTERFACE REQUIREMENTS 43

3.2 SOFTWARE INTERFACE REQUIREMENTS 49

CHAPTER 4 OTHER FUNCTIONAL REQUIREMENTS 51-52

4.1 PERFORMANCE REQUIREMENTS 51

4.2 SCALABILITY REQUIREMENTS 51

4.3 TRAINING REQUIREMENTS 51

CHAPTER 5 REFERENCES 53

LIST OF FIGURES PAGE.NO

FIGURE 1 Big Data Insights ------------------------------------------ 10

FIGURE 2 Scope of the project ---------------------------------------- 14

FIGURE 3 Jupyter Toolkit ------------ 16

FIGURE 4 Model of the project 18

FIGURE 5 Flow of data preparation 19

FIGURE 6 Preprocessing perspective 20

FIGURE 7 Tokenization Model 22

FIGURE 8 Data Frame in Pandas 25

FIGURE 9 Data Frame Content 26

FIGURE 10 Feature Selection Model 27

FIGURE 11 Count Vectors as Features 29

FIGURE 12 Word-Level Tf-Idf Features 30

FIGURE 13 N-Gram Tf-Idf Features 31

FIGURE 14 Word Embedding Model 33

FIGURE 15 Data Model Selection 34

FIGURE 16 Types of Machine Learning Algorithm 35

FIGURE 17 Decision Tree 36

FIGURE 18 K-Means clustering algorithm 37 -

FIGURE 19 Frequency Distribution 39

FIGURE 20 Flow Chart of Naive-Bayes Classification 42

FIGURE 21 Jupyter Notebook Dashboard 44

FIGURE 22 Kernal Dropdown Selection 45

FIGURE 23 Running Directories 46

FIGURE 24 Components of Notebook Dashboard 47

FIGURE 25 Kernal Software Interface 50

FIGURE 26 Client\_Server behaviour of software 51

### INTRODUCTION

* 1. PURPOSE

Humans have been generating the data for thousands of years. More recently we have seen an amazing progression in the amount of data produced from the advent of mainframes to client server to ERP and now everything is digital. For years overwhelming amount of data produced was deemless. As the importance and value of the data to an enterprise became evident, so did the proliferation of data silos within the enterprise. The ubiquity of the internet has dramatically changed the way the enterprise works. Essentially most every business became a “digital” business. The result was a data explosion. New application paradigms such as web 2.0, social media applications, cloud computing, and software-as-a-service applications further contributed to the data explosion. These new application paradigms added several new dimensions to the very definition of data. Data sources for an enterprise were no longer confined to data stores within the corporate firewalls but also to what is available outside the firewalls. Companies such as LinkedIn, Facebook, Twitter, and Netflix took advantage of these newer data sources to launch innovative product offerings to millions of end users; a new business paradigm of “consumerism” was born. Data regardless of type, location, and source increasingly has become a core business asset for an enterprise and is now categorized as belonging to two camps: internal data (enterprise application data) and external data (e.g., web data). With that, a new term has emerged: big data. So, what is the definition of this all-encompassing arena called “big data”? To start with, the definition of big data veers into 3Vs (exploding data volumes, data getting generated at high velocity and data now offering more variety).

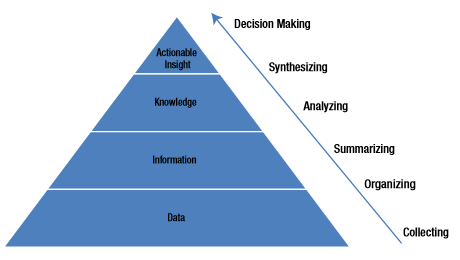


FIGURE 1: BIG DATA INSIGHTS

Operational analytics are to deliver actionable intelligence on meaningful operational metrics in real or near-real time. The realm of operational analytics is in the machine-generated data and machine-to-machine interaction data. Companies (particularly in sectors like telecommunications, logistics, transport, retailing, and manufacturing) are producing real-time operational reporting and analytics based on such data and significantly improving agility, operational visibility, and day-to-day decision making. The main underlying zeal of the analytics is derived by the use of machine learning algorithms. In the present internet advertising business paradigm, the preference of different people viewing the content while browsing can be used to classify them in pre-defined groups and send them targeted messages which are likely to imply impact on them. For this the content represented to the viewer has to be categorised provided the content can be of different formats (photos, videos, files, spreadsheets, multimedia, records, etc). Different classification methods can be used to categorize the contents of different types. Considering the text we use text classification followed by the machine learning algorithm to achieve our final goal. Text classification is used to categorize the raw data into specific groups. Here we are considering the domain of news. We collect the datasets containing headlines, URLs, reviews. Existing news portals on the WWW aim to provide users with numerous articles that are categorised into specific topics. Such a categorisation procedure improves presentation information to end user. In order to provide the advertisers, the advantage of the micro-customer behaviour segmentation.

1.2 INTENDED AUDIENCE

Website portals and the government surveys can use this method to classify the data draw the useful insights. Text classification technology is used to automatically discover, extract, and summarize the context behind unstructured content. It helps in discovering sentiments and opinions and polarity analysis concerning everything from ideas and issues to people, products, and companies. The task entails collecting data from select web sources (industry sites, the media, blogs, forums, social networks, etc.), cross-referencing this content with target entities represented in internal systems (services, products, people, programs, etc.), and extracting and summarizing the conclusions expressed in this cross-referenced content. Companies have started leveraging sentiment analysis technology to understand the voice of consumers and take timely actions such as the ones specified below:

* Monitoring and managing public perceptions of an issue, brand, organization, etc. (called reputation monitoring).
* Analyzing reception of a new or revamped service or products.
* Anticipating and responding to potential quality, pricing, or compliance issues.
* Identifying nascent market growth opportunities and trends in customer demand

Across industries, organizations are assessing ways and means to make better business decisions utilizing such untapped and plentiful information. That means as the big-data technologies evolve and more and more business use cases come into the fray, the need for ground breaking new approaches to computing, both in hardware and software, are needed. As enterprises look to innovate at a faster pace, launching innovative products and improve customer services, they need to find better ways of managing and utilizing data both within the internal and external firewalls. Organizations are realizing the need for and the importance of scaling up their existing data management practices and adopting newer information management paradigms to combat the perceived risk of reduced business insight (while the volume of data is increasing rapidly, it is also posing an interesting problem). So an organization’s ability to analyze that data to find meaningful insights is becoming increasingly complex.

1.3 PROJECT SCOPE

This method is used in data mining to make the computer understand the normal high level language and then draw out conclusions from them. It is the breakthrough technology which is used to either perform sentiment analysis or to know the anomalies in trends. It uses the machine learning techniques, natural language processing techniques to achieve the goals. It can be used in the following aspects

* + 1. Understanding audience sentiment from social media.
    2. Detection of spam and non-spam emails.
    3. Auto tagging of customer queries.

This method age has enabled enterprises of all sizes ranging from startups to small business and established large enterprises to utilize a new generation of processes and technologies. In many instances the promise of overcoming the scalability and agility challenges of traditional data management, coupled with the creative usage of data from multiple sources, have enterprise stakeholders taking serious notice of their big data potential. McKinsey’s analysis indicates that big data has the potential to add value across all industry segments. Companies likely to get the most out of big data analytics include:

Financial services: Capital markets generate large quantities of stock market and banking transaction data that can help in fraud detection, maximizing successful trades, etc.

Supply chain, logistics, and manufacturing: With RFID sensors, handheld scanners, and on-board GPS vehicle and shipment tracking, logistics and manufacturing operations produce vast quantities of information to aid in route optimization, cost savings, and operational efficiency.

Online services and web analytics: Firms can greatly benefit from increasing their customer intelligence and using it for effective cross-selling/up.

Energy and utilities: “Smart grids” and electronic sensors attached to machinery, oil pipelines and equipment generate streams of incoming data that can be used for preventive means to avoid disastrous failures.

Media and telecommunications: Streaming media, smartphones, tablets, browsing behavior and text messages aid in analyzing the user interests and behavior and improve customer retention and avoid churn.

Health care and life sciences: Analyzing electronic medical records systems in aiding optimum patient treatment options and analyzing data for clinical studies can heavily influence both individual patients’ care and public health management and policy.

Retail and consumer products: Retailers can analyze vast quantities of sales transaction data and understand the buying behaviors, as well as make effective individual-focused customized campaigns by analyzing social networking data.

Business, which can lead to enhanced productivity, a stronger competitive position, and greater innovation—all of which can have a significant impact on the bottom line. For example, collecting sensor data through in-home health-care monitoring devices can help analyze patients’ health and vital statistics proactively. This is especially critical in case of elderly patients. Health-care companies and medical insurance companies can then make time interventions to save lives or prevent expenses by reducing hospital admissions costs. The proliferation of smart phones and other GPS devices offers advertisers an opportunity to target consumers when they are in close proximity to a store, a coffee shop, or a restaurant. This opens up new revenue for service providers and offers many businesses a chance to target new customers. Retailers usually know who buys their products. Use of social media networks and web-log files from their e-commerce sites can help them understand who didn’t buy and why they chose not to. This can enable much more effective micro customer segmentation and targeted marketing campaigns, as well as improve supply chain efficiencies. Companies can now use sophisticated metrics to better understand their customers. To better manage and analyze customer information, companies can create a single source for all customer interactions and transactions. Forrester believes that organizations can maximize the value of social technologies by taking a 720-degree view of their customers instead of the previous 360-degree view. In the telecom industry, applying predictive models to manage customer churn has long been known as a significant innovation; however, today the telecom companies are exploring new data sources like customers’ social profiles to further understand customer behavior and perform micro-segmentations of their customer base. Companies must manage and analyze their customers’ profiles to better understand their interactions with their networks of friends, family, peers, and partners. For example, using social relationships the company can further analyze whether customer attrition from their customer base is also influencing similar behavior from a host of other customers who have social connections with the same customer. By doing this kind of linkage analysis companies can better target their retention campaigns and increase their revenue and profit.



FIGURE: 2 SCOPE OF THE PROJECT

1.4 OPERATING ENVIRONMENT

The ANACONDA toolkit is used to implement the text classification technique. This tool kit has a set of tools to analyse next generation sequencing data and in particular to discover and genotype insertions from paired-end reads. It aims to simplify Package management and deployement.Package versions are managed system named conda. The anaconda toolkit interface for user for developing the programming is done by ANACONDA NAVIGATOR. It is a desktop graphical user interface(GUI), included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without Command Line Commands. Navigator can search for packages on Anaconda Repository or on Anaconda cloud, install them in local environment, run the packages and update them. It is available for Windows, MacOS and Linux.

The following applications are available im Navigator

JupyterLab

Jupyter Notebook

QtConsole

Spyder

Glueviz

Orange

Rstdio

Visual Studio Code

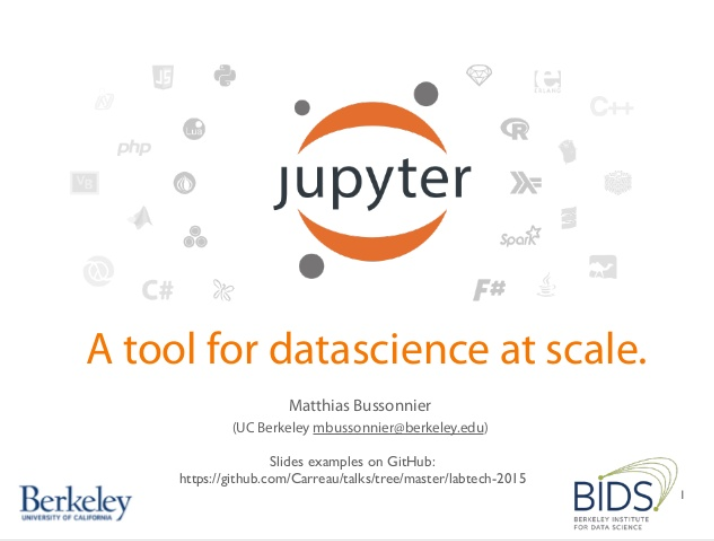


FIGURE: 3 JUPYTER TOOLKIT

We use Jupyter Notebook to implement our project.

1.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

Our project uses the Naïve Bayes algorithm to classify the documents for performing news categorization. Naive Bayes is a simple classification method based on probabilistic induction using Bayes theorem to assign class labels to test tuples. An important assumption made is that the attributes are conditionally independent. Any number of attributes or classes can be handled efficiently. The Bayesian classification allows us to include our prior knowledge in the classification process, which is hence practical. The ‘e1070’ library package from the CRAN repository is used for this purpose. The detailed analysis of the method’s accuracy is shown in the following section.

ANALYSIS OF THE METHODS

Confusion matrices are constructed for better understanding. It is a table which consists of information about predictions made by a classification system.

TP - True Positive is the number of correct predictions that an instance is positive

TN - True Negative is the number of correct predictions that an instance is Negative

FP - False Positive is the number of incorrect predictions that an instance is positive

FN - False Negative is the number of incorrect predictions that an instance is negative

Accuracy - It is the ratio of the true results (Both true positives and true negatives) to the total number of instances examined.

Accuracy = ( TP+TN ) / ( TP+TN+FP+FN )

Precision - Precision is the ratio of the true positive to all the positive values (Both true and false positives)

Precision = TP / ( TP+FP )

Accuracy refers to the closeness of a measured value to a known value whereas precision refers to the closeness of two measurements to each other.

Recall - It is the ratio of true positive to actually positive values, It gives information about when it is actually yes and how often does it predict yes.

Recall = TP / ( FN+TP )

Precision is how many of the found were correct hits and recall is how many of the correct hits were also found.

2. SYSTEM FEATURES

One of the widely used natural language processing task in different business problems is “Text Classification”. The goal of text classification is to automatically classify the text documents into one or more defined categories.

Text Classification is an example of supervised machine learning task since a labelled dataset containing text documents and their labels is used for train a classifier. An end-to-end text classification pipeline is composed of three main components:

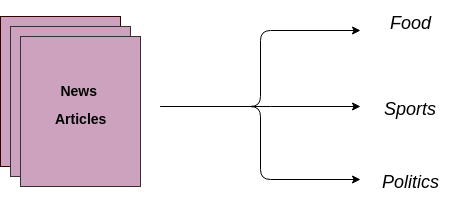


FIGURE: 4 MODEL OF PROJECT

1. Dataset Preparation: The first step is the Dataset Preparation step which includes the process of loading a dataset and performing basic pre-processing. The dataset is then splitted into train and validation sets.

2. Feature Engineering: The next step is the Feature Engineering in which the raw dataset is transformed into flat features which can be used in a machine learning model. This step also includes the process of creating new features from the existing data.

3. Model Training: The final step is the Model Building step in which a machine learning algorithm of Text Classifier on a labelled dataset.

1. DATA SET PREPARATION

The raw data is processed through the sequence of steps so that the operation of the machine learning algorithms is logically efficient. Different types of the data with different machine learning algorithms use several combinations of preprocessing steps. In the area of Text Mining, data preprocessing used for extracting interesting and non-trivial and knowledge from unstructured text data. Information Retrieval (IR) is essentially a matter of deciding which documents in a collection should be retrieved to satisfy a user's need for information. The user's need for information is represented by a query or profile, and contains one or more search terms, plus some additional information such as weight of the words. Hence, the retrieval decision is made by comparing the terms of the query with the index terms (important words or phrases) appearing in the document itself. The decision may be binary (retrieve/reject), or it may involve estimating the degree of relevance that the document has to query.

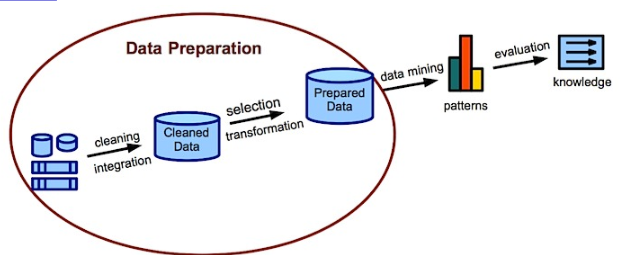


FIGURE 5: FLOW OF DATA PREPARATION

Unfortunately, the words that appear in documents and in queries often have many structural variants. So before the information retrieval from the documents, the data preprocessing techniques are applied on the target data set to reduce the size of the data set which will increase the effectiveness of IR system. The important steps used are

a.Tokenisation.

b.Stop word removal .

c.Stemming for the text documents.

Text pre-processing is an essential part of any NLP system, since the characters, words, and sentences identified at this stage are the fundamental units passed to all further processing stages, from analysis and tagging components, such as morphological analyzers and part-of speech taggers, through applications, such as information retrieval and machine translation systems. It is a Collection of activities in which Text Documents are pre-processed. Because the text data often contains some special formats like number formats, date formats and the most common words that unlikely to help Text mining such as prepositions, articles, and pro-nouns can be eliminated.

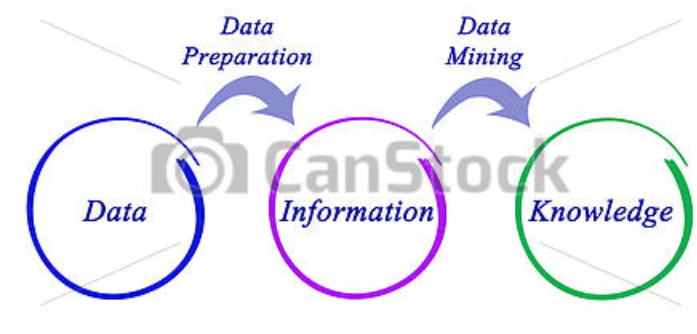


FIGURE 6: PREPROCESSING PERSPECTIVE

Need of Text Preprocessing in NLP System

* To reduce indexing(or data) file size of the Text documents
  + Stop words accounts 20-30% of total word counts in a particular text documents.
  + Stemming may reduce indexing size as much as 4050%.
* To improve the efficiency and effectiveness of the IR system
  + Stop words are not useful for searching or Text mining and they may confuse the retrieval system.
  + Stemming used for matching the similar words in a text document

1. Tokenization

Tokenization is the process of breaking a stream of text into words, phrases, symbols, or other meaningful elements called tokens .The aim of the tokenization is the exploration of the words in a sentence. The list of tokens becomes input for further processing such as parsing or text mining. Tokenization is useful both in linguistics (where it is a form of text segmentation), and in computer science, where it forms part of lexical analysis.

Textual data is only a block of characters at the beginning. All processes in information retrieval require the words of the data set. Hence, the requirement for a parser is a tokenization of documents. This may sound trivial as the text is already stored in machine-readable formats. Nevertheless, some problems are still left, like the removal of punctuation marks. Other characters like brackets, hyphens, etc require processing as well. Furthermore, tokenizer can cater for consistency in the documents. The main use of tokenization is identifying the meaningful keywords. The inconsistency can be different number and time formats. Another problem are abbreviations and acronyms which have to be transformed into a standard form.

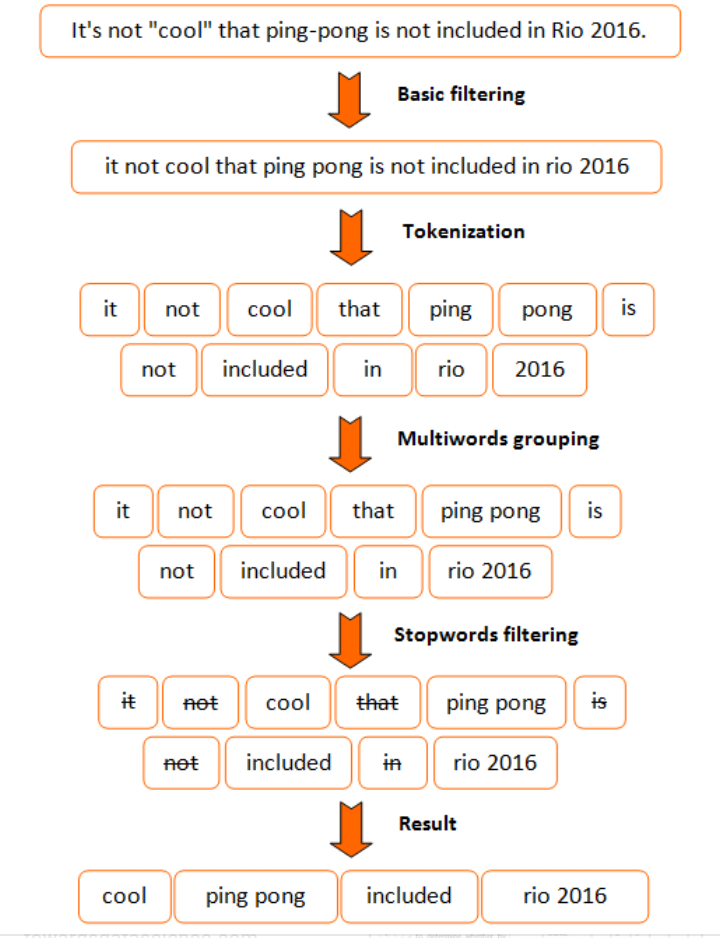


FIGURE 7: TOKENISATION MODEL

Challenges in Tokenization

Challenges in tokenization depend on the type of language. Languages such as English and French are referred to as space delimited as most of the words are separated from each other by white spaces. Languages such as Chinese and Thai are referred to as unsegmented as words do not have clear boundaries. Tokenizing unsegmented language sentences requires additional lexical and morphological information. Tokenization is also affected by writing system and the typographical structure of the words. Structure of languages can be grouped into three categories:

* Isolating: Words do not divide into smaller units. Example: Mandarin Chinese
* Agglutinative: Words divide into smaller units. Example: Japanese, Tamil
* Inflectional: Boundaries between morphemes are not clear and ambiguous in terms of grammatical meaning. Example: Latin.

1. STOP WORD REMOVAL

Many words in documents recur very frequently but are essentially meaningless as they are used to join words together in a sentence. It is commonly understood that stop words do not contribute to the context or content of textual documents. Due to their high frequency of occurrence, their presence in text mining presents an obstacle in understanding the content of the documents.

Stop words are very frequently used common words like ‘and’, ‘are’, ‘this’ etc. They are not useful in classification of documents. So they must be removed. However, the development of such stop words list is difficult and inconsistent between textual sources. This process also reduces the text data and improves the system performance. Every text document deals with these words which are not necessary for text mining applications.

1. Stemming

Stemming is the process of conflating the variant forms of a word into a common representation, the stem. For example, the words: “presentation”, “presented”, “presenting” could all be reduced to a common representation “present”.

This is a widely used procedure in text processing for information retrieval (IR) based on the assumption that posing a query with the term presenting implies an interest in documents containing the words presentation and presented.

Errors in Stemming

There are mainly two errors in stemming.

1. Over stemming.

2. Under stemming .

Over-stemming is when two words with different stems are stemmed to the same root. This is also known as a false positive. Under-stemming is when two words that should be stemmed to the same root are not. This is also known as a false negative.

TYPES OF STEMMING ALGORITHMS

i) Table Look Up Approach

One method to do stemming is to store a table of all index terms and their stems. Terms from the queries and indexes could then be stemmed via lookup table, using btrees or hash tables. Such lookups are very fast, but there are problems with this approach. First there is no such data for English, even if there were they may not be represented because they are domain specific and require some other stemming methods. Second issue is storage overhead.

ii) Affix Removal Stemmers

Affix removal stemmers removes the suffixes or prefixes from the terms leaving the stem. One of the example of the affix removal stemmer is one which removes the plurals form of the terms. Some set of rules for such a stemmer are as follows (Harman) a) If a word ends in “ies” but not “eies” or “aies ” Then “ies” -> “y” b) If a word ends in “es” but not “aes”, or “ees” or “oes” Then “es” -> “e” c) If a word ends in “s” but not “us” or “ss ” Then “s” -> “NULL”

These pre-processing techniques eliminates noisy from text data, later identifies the root word for actual words and reduces the size of the text data. This improves performance of the IR system.

1. Dataset preparation

To prepare the dataset, load the downloaded data into a pandas dataframe containing two columns – text and label. (Source)

# load the dataset

data = open('data/corpus').read()

labels, texts = [], []

for i, line in enumerate(data.split("\n")):

content = line.split()

labels.append(content[0])

texts.append(" ".join(content[1:]))

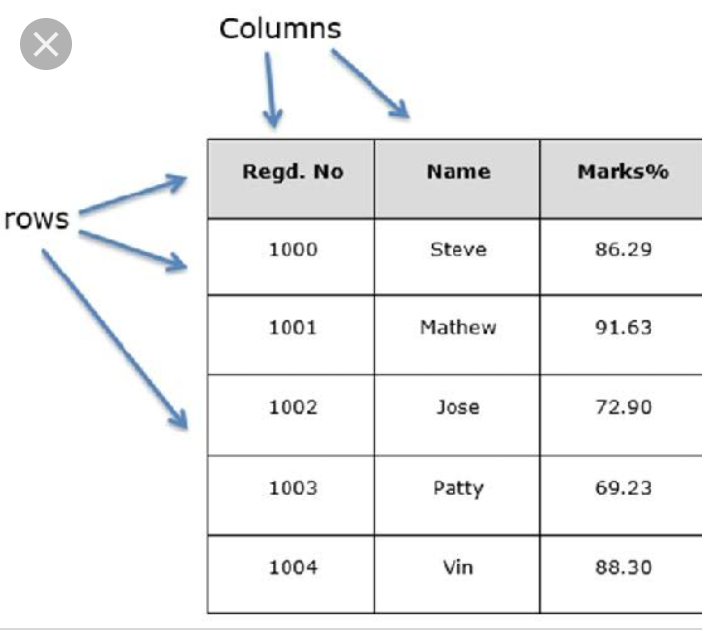


FIGURE 8: DATAFRAME IN PANDAS

# create a dataframe using texts and lables

trainDF = pandas.DataFrame()

trainDF['text'] = texts

trainDF['label'] = labels

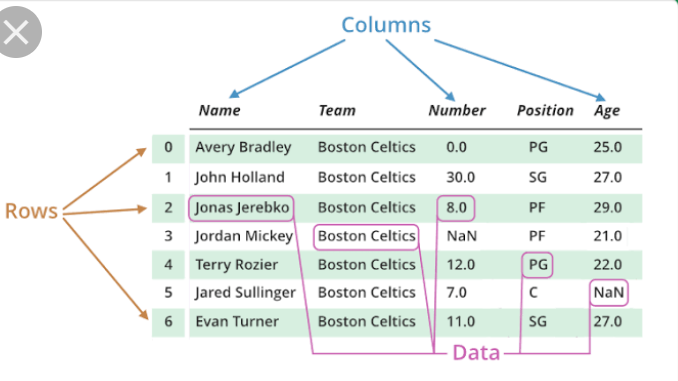


FIGURE 9: DATAFRAME CONTENT

Next, we will split the dataset into training and validation sets so that we can train and test classifier. Also, we will encode our target column so that it can be used in machine learning models.

# split the dataset into training and validation datasets

train\_x,valid\_x,train\_y,valid\_y= model\_selection.train\_test\_split(trainDF['text'], trainDF['label'])

# label encode the target variable

encoder = preprocessing.LabelEncoder()

train\_y = encoder.fit\_transform(train\_y)

valid\_y = encoder.fit\_transform(valid\_y)

1. Feature Engineering

The next step is the feature engineering step. In this step, raw text data will be transformed into feature vectors and new features will be created using the existing dataset. We will implement the following different ideas in order to obtain relevant features from our dataset.

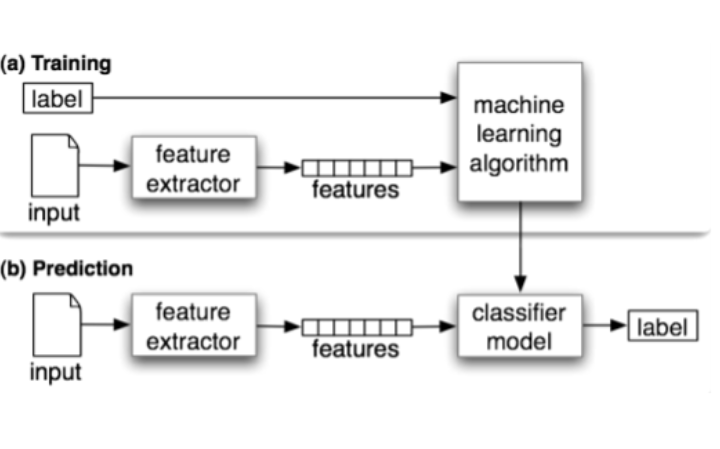
FIFF

FIGURE 10: FEATURE SELECTION MODEL

1. Count Vectors as features
2. TF-IDF Vectors as features

Word level

N-Gram level

Character level

1. Word Embeddings as features
2. Text / NLP based features
3. Count Vectors as features

Count Vector is a matrix notation of the dataset in which every row represents a document from the corpus, every column represents a term from the corpus, and every cell represents the frequency count of a particular term in a particular document.

# create a count vectorizer object

count\_vect = CountVectorizer(analyzer='word', token\_pattern=r'\w{1,}')

count\_vect.fit(trainDF['text'])

# transform the training and validation data using count vectorizer object

xtrain\_count = count\_vect.transform(train\_x)

xvalid\_count = count\_vect.transform(valid\_x)

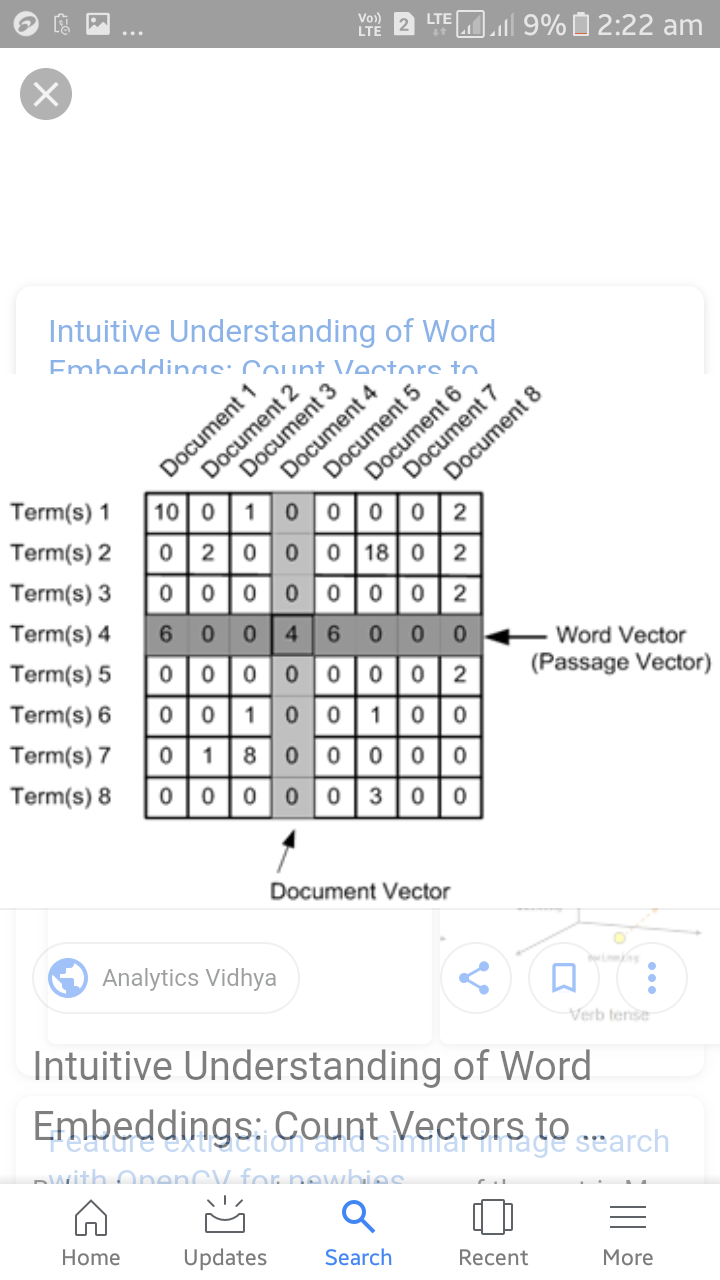


FIGURE 11 :COUNT VECTORS AS FEATURES.

1. TF-IDF Vectors as features

TF-IDF score represents the relative importance of a term in the document and the entire corpus. TF-IDF score is composed by two terms: the first computes the normalized Term Frequency (TF), 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(t) = (Number of times term t appears in a document) / (Total number of terms in the document)

IDF(t) = log\_e(Total number of documents / Number of documents with term t in it)

TF-IDF Vectors can be generated at different levels of input tokens (words, characters, n-grams)

* Word Level TF-IDF : Matrix representing tf-idf scores of every term in different documents.

# word level tf-idf

tfidf\_vect = TfidfVectorizer(analyzer='word', token\_pattern=r'\w{1,}', max\_features=5000)

tfidf\_vect.fit(trainDF['text'])

xtrain\_tfidf = tfidf\_vect.transform(train\_x)

xvalid\_tfidf = tfidf\_vect.transform(valid\_x)

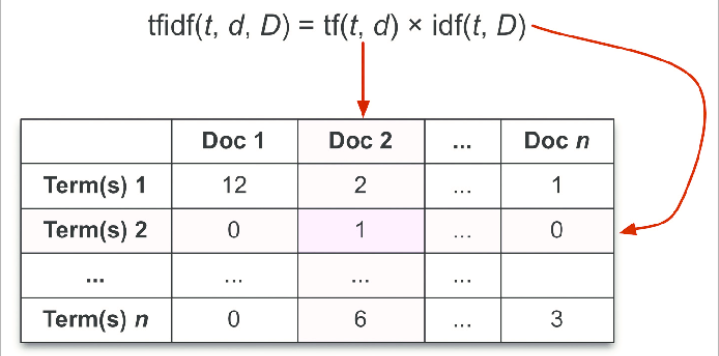


FIGURE 12: WORD-LEVEL TF-IDF FEATURES

* N-gram Level TF-IDF : N-grams are the combination of N terms together. This Matrix representing tf-idf scores of N-grams.

# ngram level tf-idf

tfidf\_vect\_ngram = TfidfVectorizer(analyzer='word', token\_pattern=r'\w{1,}', ngram\_range=(2,3), max\_features=5000)

tfidf\_vect\_ngram.fit(trainDF['text'])

xtrain\_tfidf\_ngram = tfidf\_vect\_ngram.transform(train\_x)

xvalid\_tfidf\_ngram = tfidf\_vect\_ngram.transform(valid\_x)

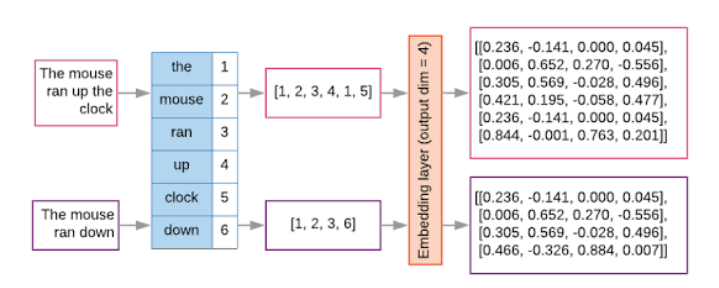


FIGURE 13: N-GRAM TF-IDF FEATURES

* Character Level TF-IDF : Matrix representing tf-idf scores of character level n-grams in the corpus

# characters level tf-idf

tfidf\_vect\_ngram\_chars = TfidfVectorizer(analyzer='char', token\_pattern=r'\w{1,}', ngram\_range=(2,3), max\_features=5000)

tfidf\_vect\_ngram\_chars.fit(trainDF['text'])

xtrain\_tfidf\_ngram\_chars = tfidf\_vect\_ngram\_chars.transform(train\_x)

xvalid\_tfidf\_ngram\_chars = tfidf\_vect\_ngram\_chars.transform(valid\_x)

c. Word Embeddings

A word embedding is a form of representing words and documents using a dense vector representation. The position of a word within the vector space is learned from text and is based on the words that surround the word when it is used. Word embeddings can be trained using the input corpus itself or can be generated using pre-trained word embeddings such as Glove, FastText, and Word2Vec.Following snnipet shows how to use pre-trained word embeddings in the model. There are four essential steps:

* Create a mapping of token and their respective embeddings
* You can download the pre-trained word embeddings from here(# load the pre-trained word-embedding Loading the pretrained word embeddings)
* Creating a tokenizer object
* Transforming text documents to sequence of tokens and pad them

embeddings\_index = {}

for i, line in enumerate(open('data/wiki-news-300d-1M.vec')):

values = line.split()

embeddings\_index[values[0]] = numpy.asarray(values[1:], dtype='float32')

# create a tokenizer

token = text.Tokenizer()

token.fit\_on\_texts(trainDF['text'])

word\_index = token.word\_index

# convert text to sequence of tokens and pad them to ensure equal length vectors

train\_seq\_x = sequence.pad\_sequences(token.texts\_to\_sequences(train\_x), maxlen=70)

valid\_seq\_x = sequence.pad\_sequences(token.texts\_to\_sequences(valid\_x), maxlen=70)

# create token-embedding mapping

embedding\_matrix = numpy.zeros((len(word\_index) + 1, 300))

for word, i in word\_index.items():

embedding\_vector = embeddings\_index.get(word)

if embedding\_vector is not None:

embedding\_matrix[i] = embedding\_vector

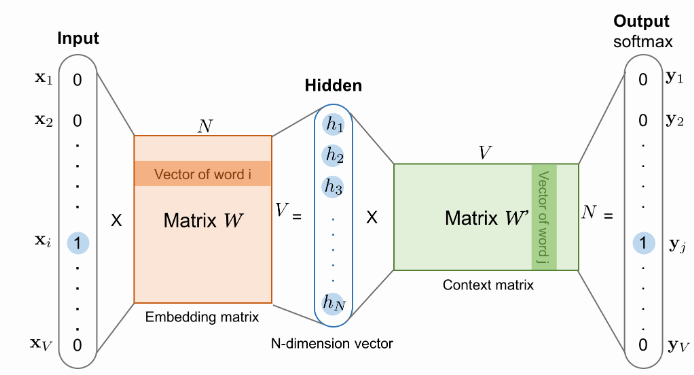


FIGURE 14 :WORD EMBEDDING MODEL

d. Text / NLP based features

A number of extra text based features can also be created which sometimes are helpful for improving text classification models. Some examples are:

Word Count of the documents – total number of words in the documents

Character Count of the document – total number of characters in the documents

Average Word Density of the documents – average length of the words used in the documents

Puncutation Count in the Complete Essay – total number of punctuation marks in the documents

Upper Case Count in the Complete Essay – total number of upper count words in the documents

Title Word Count in the Complete Essay – total number of proper case (title) words in the documents

Frequency distribution of Part of Speech Tags:

Noun Count

Verb Count

Adjective Count

Adverb Count

Pronoun Count

These features are highly experimental ones and should be used according to the problem statement only.

3 Model Building

Here we are building the model “text.clr”.

Generally model building is commenced after model selection.this model selection is the process of

Choosing between different machine learning approaches-SVM, logistic regression, ,supervised or unsupervised.

Choosing between different set of hyperparameters or sets of features-decision of polynomial degrees or complexities for the above selected machine learning approaches.

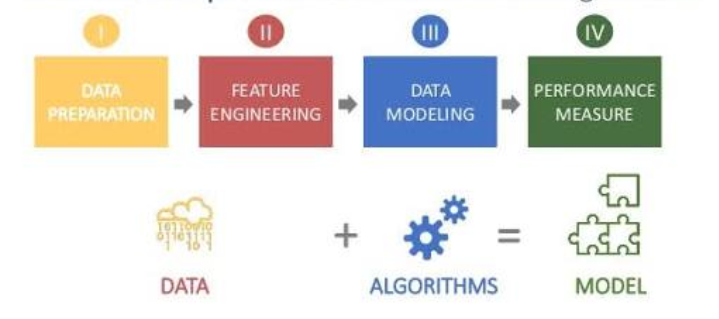


FIGURE 15 : DATA MODEL SELECTION

There are many different choices of machine learning models which can be used to train a final model. There exist numerous types of machine-learning algorithms. Some of the more popular approaches include supervised learning, unsupervised learning, and probabilistic learning.

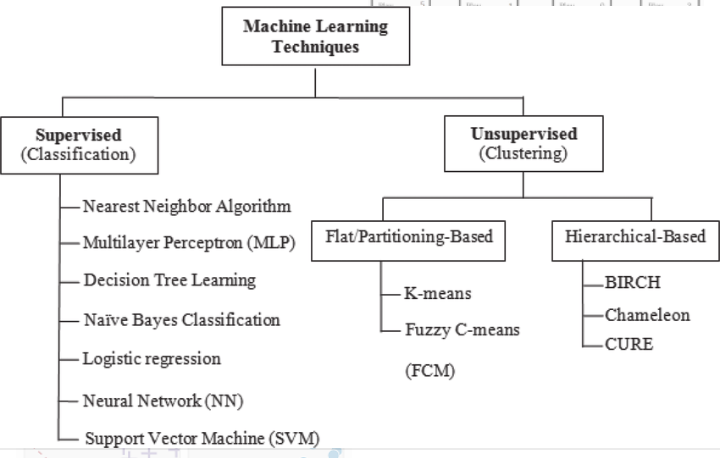


FIGURE 16: TYPES OF MACHINE LEARNIN APPROACHES

SUPERVISED LEARNING: Supervised learning algorithms imply that a teacher is present to identify when a result is right or wrong. The input data contains both a predictor (independent variable) and target (dependent variable) whose value is to be estimated. Through the process of supervised learning, the algorithm predicts the value of the target variable based on the predictor variables of supervised learning algorithms include perceptrons, backpropagation, and decision trees.

Supervised Learning Supervised learning algorithms use training data that has been classiﬁed (has a target value for each training vector). The purpose of the supervised learning algorithm is to create a prediction function using the training data that will generalize for unseen training vectors to classify them correctly.Following, decision-tree learning is explored to construct a decision tree from observed behaviour.

Learning with Decision Trees: One of the most intuitive and practical methods for supervised learning is the decision tree. A decision tree is the result of a classiﬁcation process in which the source data is reduced into a predictor tree that represents a set of if/then/else rules. As an example, consider an observer that watched a FPS player and recorded their actions when confronted with an enemy. The result is tabulated as follows

Table :Observed actions of a player in a ﬁrst-person-shooter game.

WEAPON AMMO HEALTH BEHAVIOUR

Gun Full Low Fight

Gun Low Full Evade

Knife Low Full Fight

Knife Low Low Evade

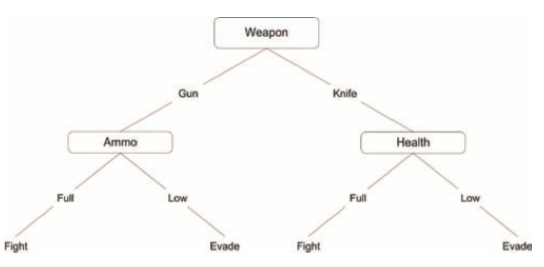


FIGURE 17 : DECISION TREE

If ( (Weapon == gun) && (Ammo ==Full) ) then Fight

Else if ( (Weapon ==knife) && (Health == Full) ) then Fight

Else Evade

As shown in Figure and Table , this player has three predictor variables. These are Weapon the weapon currently carried by the player), Ammo (the amount of ammunition carried by the player), and ﬁnally, Health (the level of health of the player). The interior nodes of the decision tree are the features, and the arcs out of the feature nodes are feature values. Each leaf in the tree is a category (or class). The breakdown of this decision tree results in a simple conditional expression shown in Figure. This condition expression (representing the decision tree) deﬁnes that if the player has a gun with full ammo, it will ﬁght. If the player has a knife and full health, it will ﬁght. Otherwise, the player will evade. This is clearly shown in the simple conditional in figure.

UNSUPERVISED LEARNING: Unsupervised learning algorithms imply that learning occurs unsupervised, or without a teacher. In this case, there is no target variable, but instead relationships in the data that are exploited for classiﬁcation (for example, patterns in the data). Examples of unsupervised learning include Hebbian Learning, Vector Quantization, and Adaptive Resonance Theory (ART).

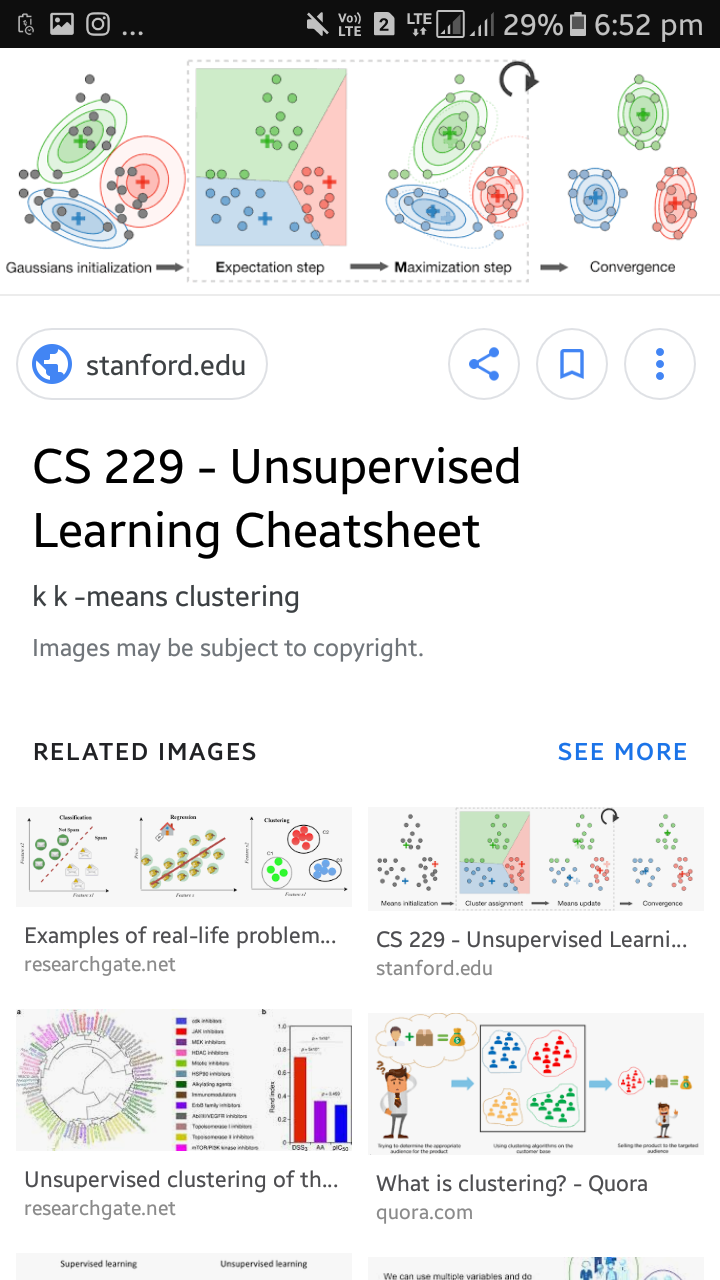


FIGURE 18 :K-MENS CLUSTERING UN SUPERVISED ALGORITHM

PROBABLISTIC NETWORK: Probabilistic approaches to learning is a useful method in modern AI. This approach works on the principle that assigning probabilities to events can be done based on prior probabilities and observed data. This is useful because in theory this method can arrive at optimal decisions.

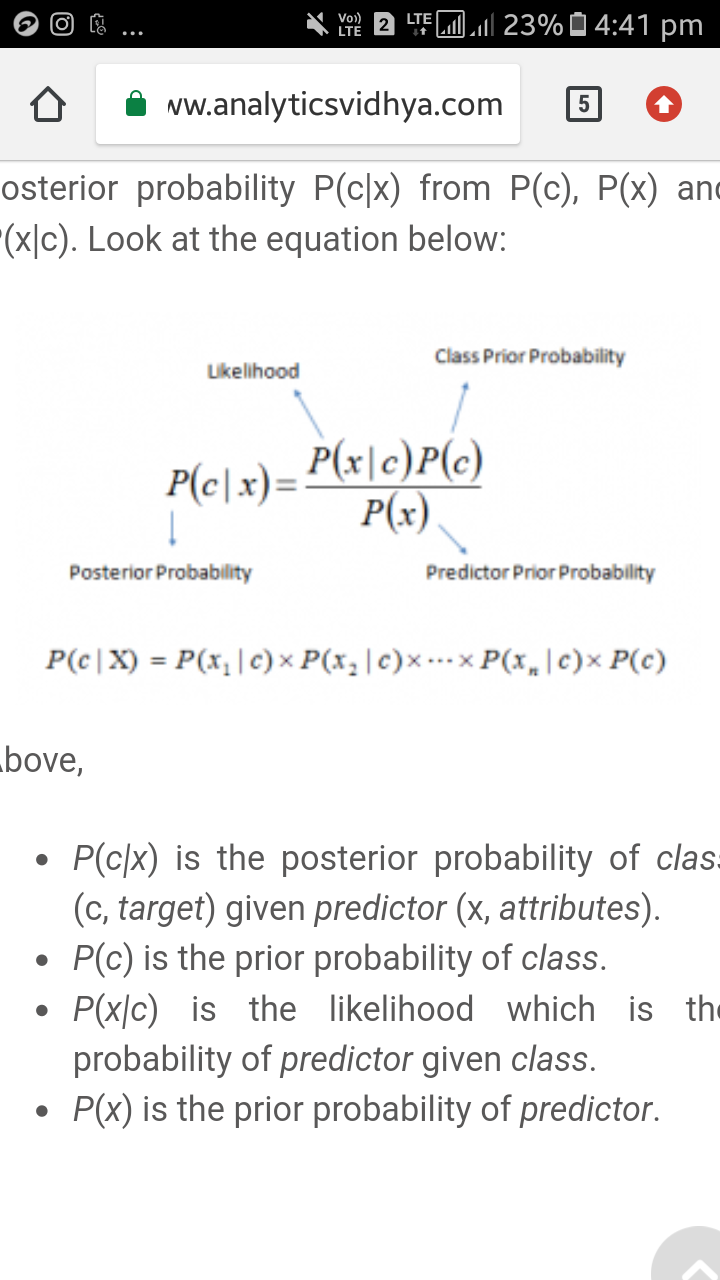
We use the supervised approach Naïve Bayes algorithm to perform the arinin and the testing of the data.

Naïve Bayes algorithm

It is a classification technique based on Bayes’ Theorem with an assumption of independence among predictors. In simple terms, a Naïve Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it to be red, round, and about 3 inches in diameter. Even if these feature depend on each other or upon the existence of the other features, all of the other properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naïve’.

Naïve Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naïve Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c/x) from p(c), p(x) and p(x/c). Look at the equation below:



Above,

* P(c/x) is the posterior probability of class (c, target) given predictor (x, attributes).
* P(c) is the prior probability of class.
* P(x/c) is the likelihood which is the probability of predictor given class.
* P(x) is the prior probability of predictor.

# How Naïve Bayes algorithm work?

Let’s understand it using an example. Below I have a training data set of the weather and corresponding target variables ‘play’ (suggesting possibilities of playing). Now, we need to classify whether players will pay or not based on weather condition. Let’s follow the below steps to perform it.

Step 1 : Convert the data set into a frequency table

Step 2: Create Likelihood table by finding the probabilities like overcast probability=0.29 and probability of playing is 0.64.

Step 3: Now, use naïve Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of the prediction.

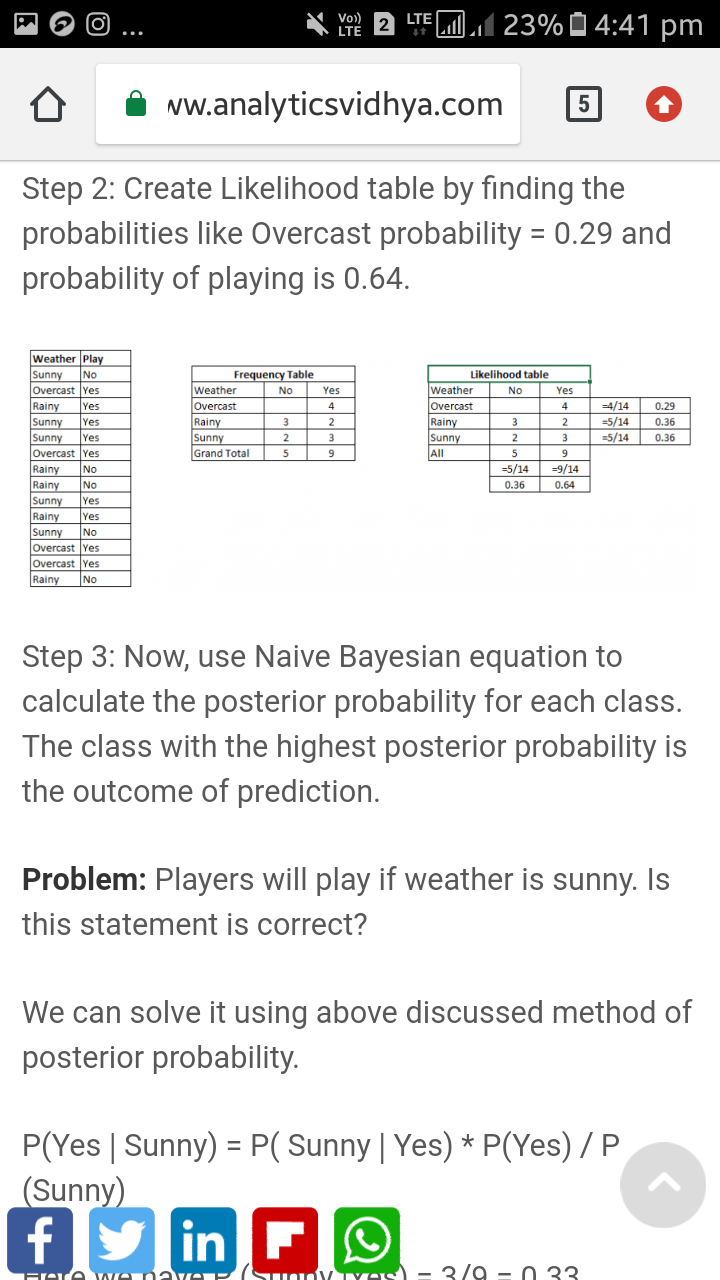


FIGURE 19: FREQUENCY DISTRIBUTION

Considering the example problem

Players will play if the weather is sunny.is this statement .

We can solve it using the above discussed method of posterior probability.

P(yes|Sunny)=P(Sunny|yes)\*P(Yes)/P(Sunny)

Here we have P(Sunny|yes)=3/9=0.33

P(Sunny)=5/14=0.36,p(yes)=9/14=0.64

Now we have P(Yes|sunny)=0.33\*0.64/0.36.

Naïve Bayes uses similar method to predict the probability of different classes based on the various attributes.

PROS OF NAÏVE BAYES CLASSIFICATION:

* It is easy and fast to predict class of test data set. It also performs well in multi class prediction.
* When assumption of independence holds, a naïve bayes classifier performs better compare to other models

CONS OF NAÏVE BAYES CLASSIFICATION :

* There is the probability of zero frequency problem due to the absence of the specified category in the training set.
* The assumption of independence predictors is not practical.

APPLICATION OF NAÏVE BAYES CLASSIFICATION:

* Real Time Prediction.
* Multi Class prediction.
* Text classification/spam Filtering/sentiment analysis.
* Recommendation system.

NAÏVE BAYES BASIC MODEL USING PYTHON

The python have many libraries to support our machine learning algorithm

GAUSSIAN: It is used in classification and it assumes that features follow a normal distribution.

MULTINOMIAL: It is used for the discrete counts like word occurring in the document. The estimate te would be how often word occurs in the document.

BERNOULLI: The binomial model is used foe feature vectors are binary, i.e either zeros or ones.

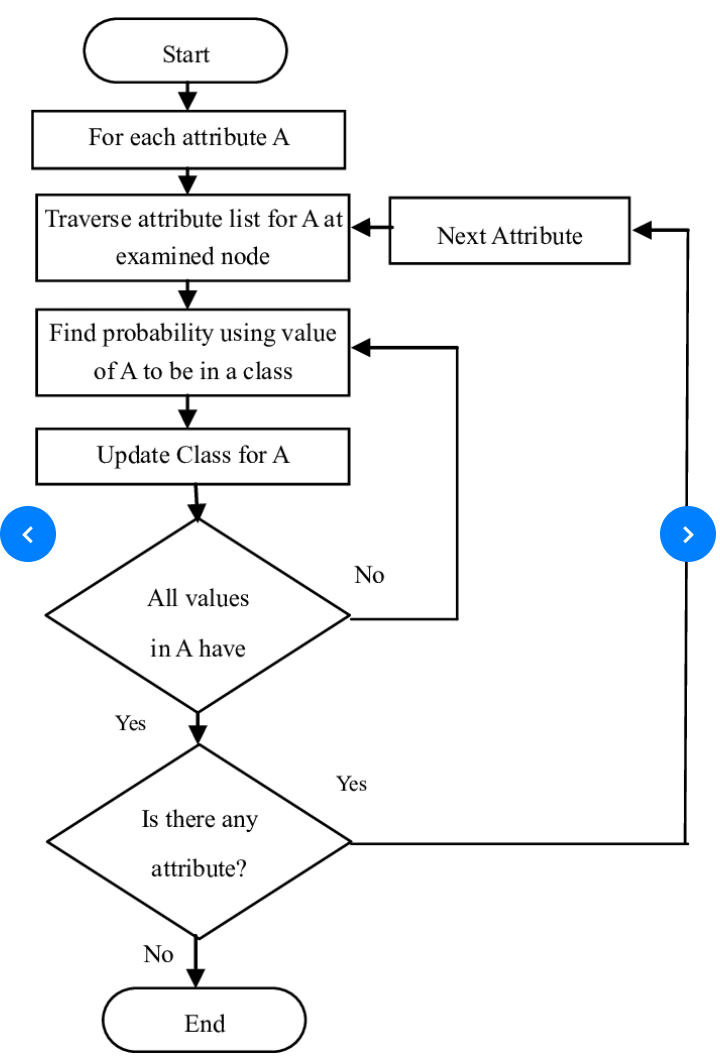


FIGURE 20: FLOW CHART OF NAÏVE BAYES CLASSIFICATION

The implementation is as followed:

The following function is a utility function which can be used to train a model. It accepts the classifier, feature\_vector of training data, labels of training data and feature vectors of valid data as inputs. Using these inputs, the model is trained and accuracy score is computed.

def train\_model(classifier, feature\_vector\_train, label, feature\_vector\_valid, is\_neural\_net=False):

# fit the training dataset on the classifier

classifier.fit(feature\_vector\_train, label)

# predict the labels on validation dataset

predictions = classifier.predict(feature\_vector\_valid)

if is\_neural\_net:

predictions = predictions.argmax(axis=-1)

return metrics.accuracy\_score(predictions, valid\_y)

# Naive Bayes on Count Vectors

accuracy = train\_model(naive\_bayes.MultinomialNB(), xtrain\_count, train\_y, xvalid\_count)

print "NB, Count Vectors: ", accuracy

# Naive Bayes on Word Level TF IDF Vectors

accuracy = train\_model(naive\_bayes.MultinomialNB(), xtrain\_tfidf, train\_y, xvalid\_tfidf)

print "NB, WordLevel TF-IDF: ", accuracy

# Naive Bayes on Ngram Level TF IDF Vectors

accuracy = train\_model(naive\_bayes.MultinomialNB(), xtrain\_tfidf\_ngram, train\_y, xvalid\_tfidf\_ngram)

print "NB, N-Gram Vectors: ", accuracy

# Naive Bayes on Character Level TF IDF Vectors

accuracy=train\_model(naive\_bayes.MultinomialNB(), xtrain\_tfidf\_ngram\_chars, train\_y, xvalid\_tfidf\_ngram\_chars)

print "NB, CharLevel Vectors: ", accuracy

### 3. EXTERNAL INTERFACE REQUIREMENTS

# USER INTERFACE

We use jupyter notebook to launch application. The note book extends the console-based approach to interactive computing in a qualitatively new direction, providing a web-based application suitable for capturing the whole computation process:

NOTEBOOK DASHBOARD:

The main GUI components are those of the notebook dashboard which is the front end of our project.

When we launch jupyter notebook, the first page we encounter is note book dashboard

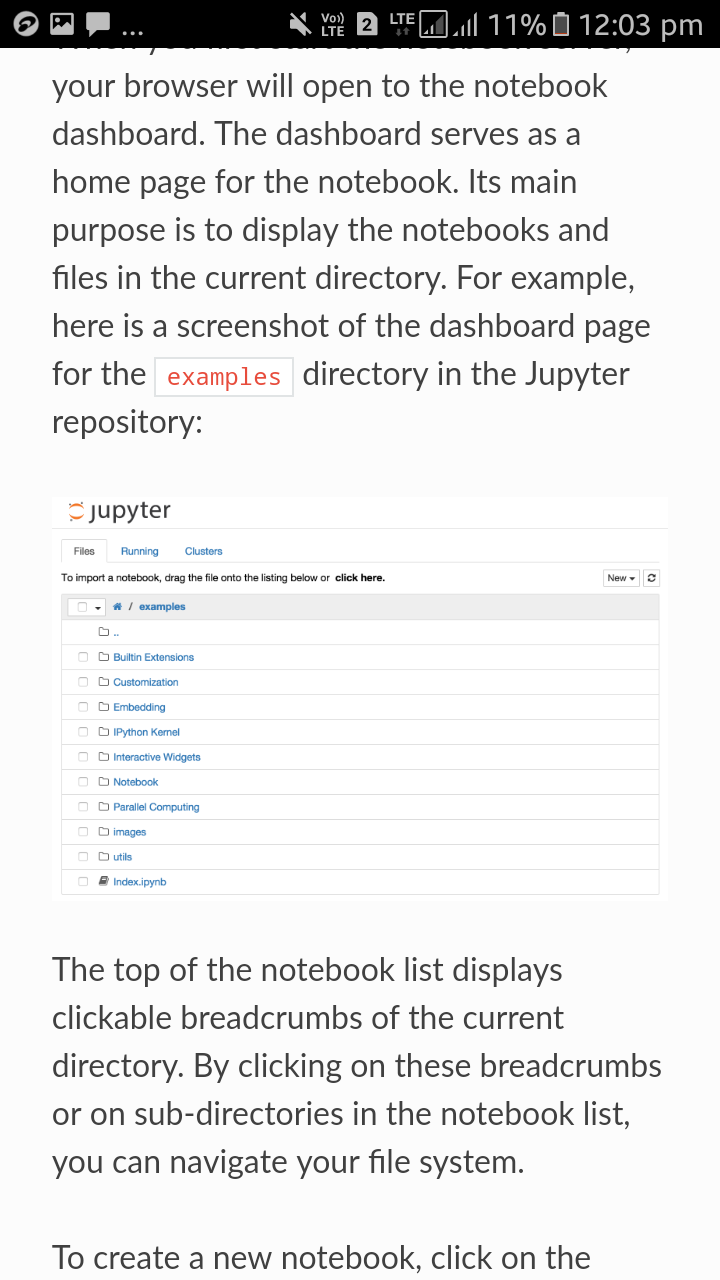


FIGURE 21 : JUPYTER NOTEBOOK DASHBOARD

Its main purpose is to display the notebooks and the files in the current directory as shown above. The top of the notebook list displays clickable breadcrumbs of the current or on sub directories in the note book list.

To create a new note book, click on the new button at the top of the list and select the kernel from the dropdown.

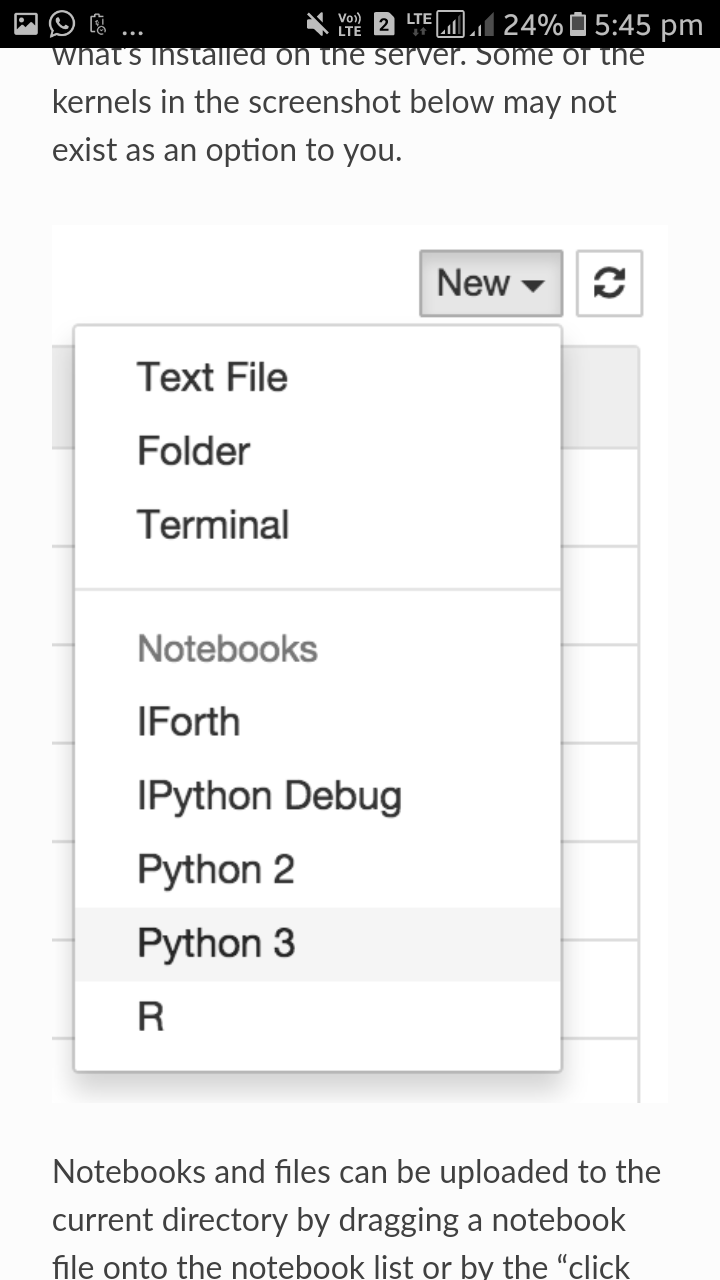


FIGURE 22: KERNEL DROPDOWN SELECTION

While running the notebook list shows green “Running text “and a green note book icon next to the running until we explicitly shut down them.

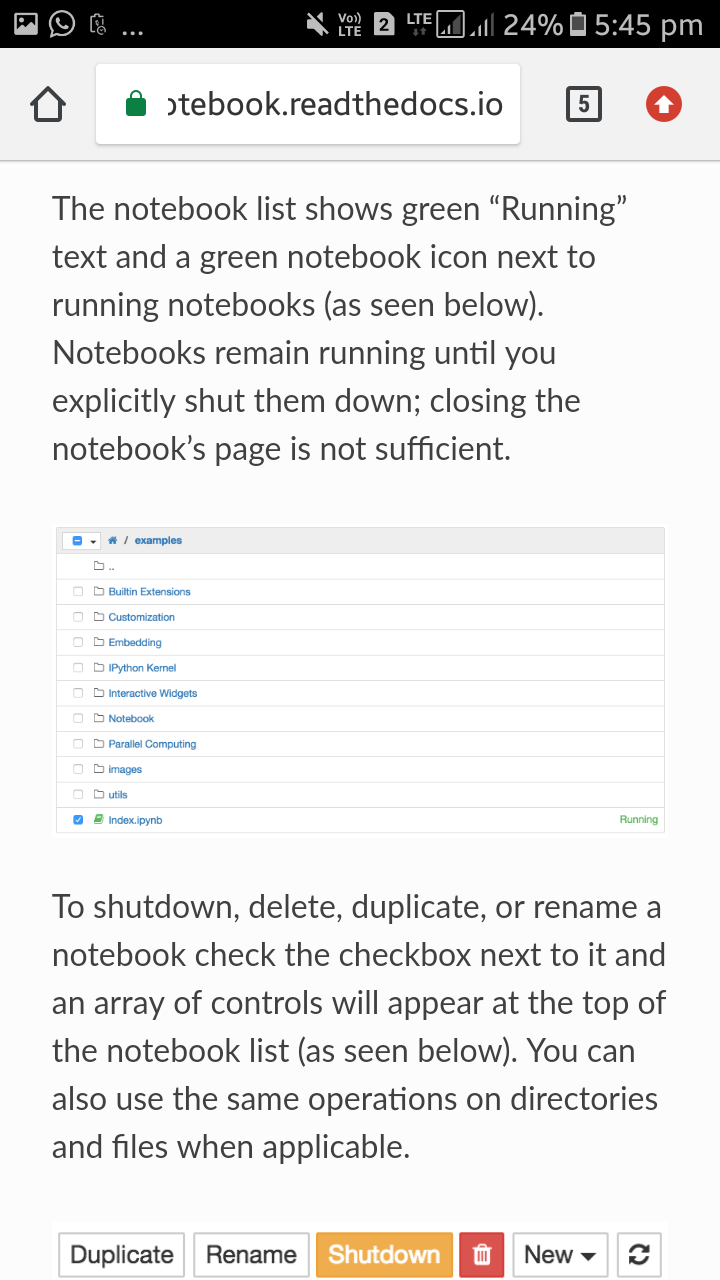
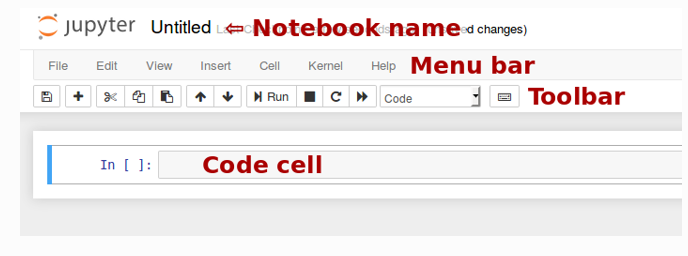


FIGURE 23: RUNNING DIRCTORIES

The notebook has the following important components

* Menu
* Toolbar
* Notebook area and cells



. FIGURE 24: COMPONENTS OF NOTEBOOK DASHBOARD.

## Starting the notebook server:

jupyter notebook

The above command can be used to start the notebook.

This will print some information about the notebook server in your console, and open a web browser to the URL of the web application (by default, http://127.0.0.1:8888).

The landing page of the Jupyter notebook web application, the **dashboard**, shows the notebooks currently available in the notebook directory (by default, the directory from which the notebook server was started).

You can create new notebooks from the dashboard with the New Notebook button, or open existing ones by clicking on their name. You can also drag and drop .ipnyb notebooks and standard .py Python source code files into the notebook list area.

When starting a notebook server from the command line, you can also open a particular notebook directly, bypassing the dashboard, with .jupyternotebook my notebook.ipynb.

When you are inside an open notebook, the *File | Open…* menu option will open the dashboard in a new browser tab, to allow you to open another notebook from the notebook directly.

### 3.2 Software interfaces

The basic workflow in a notebook is similar to standard Ipython session, you obtain desired results, rather than having to rerun separate scripts with %run command.

Typically, you will work on the computational problems in pieces, oranising related ideas into cells and moving forward once the previous parts work correctly. This is much more convenient for interactive exploration than breaking up the computation in sequential manner.

This functionality is given in the notebook by three components:

THE NOTEBOOK WEB APPLICATION: An interactive web application for writing and running code interactively.

KERNALS: Separate processes started by the notebook web application that runs users code, also handles things like computation for interactive widgets, tab completion, and introspection.

NOTEBOOK DOCUMENTS: Self-contained documents that contain a representation of all content visible in the notebook web application, including inputs and outputs of the computations, narrative text, equations ,images and rich media representations of objects. Each notebook document has its own kernel.

Notebook documents contains the inputs and outputs of a interactive session as well as additional text that accompanies the code but is not meant for execution. In this way, notebook files can serve as a complete computational record of a session, interleaving executable code with explanatory text, mathematics, and rich representations of resulting objects. These documents are internally [JSON](https://en.wikipedia.org/wiki/JSON) files and are saved with the .ipynd extension. Since JSON is a plain text format, they can be version-controlled and shared with colleagues

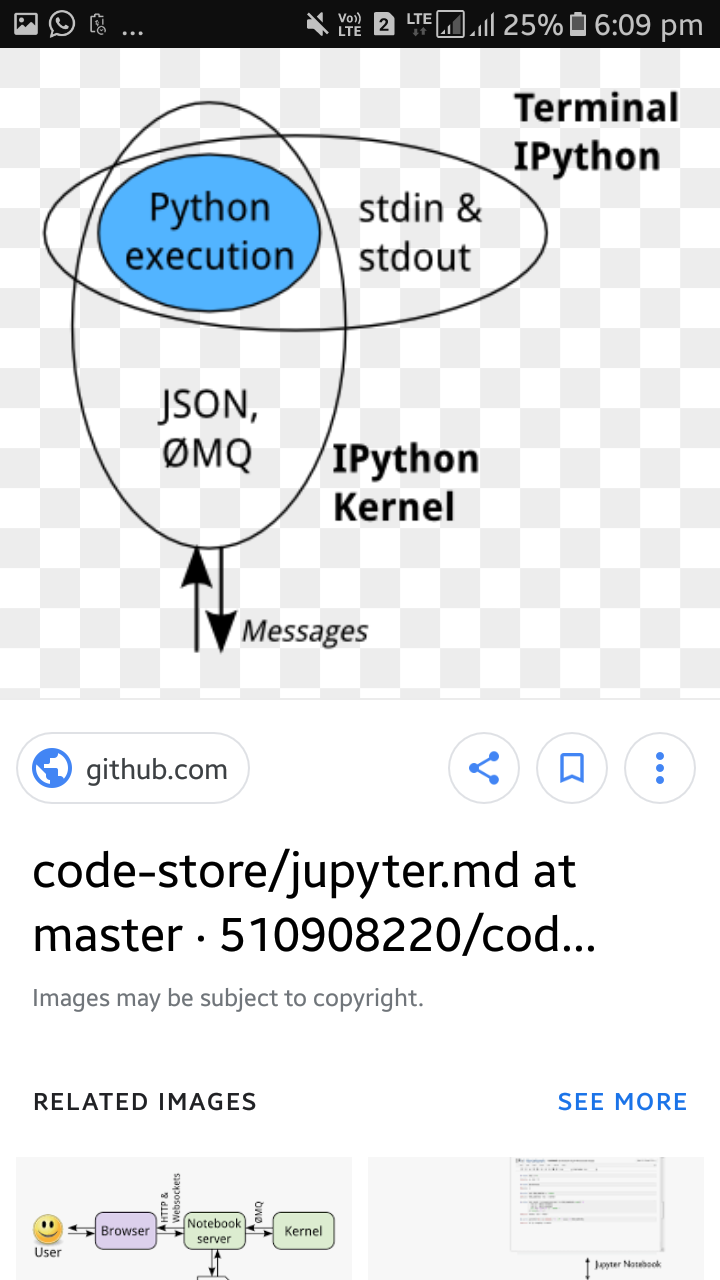


FIGURE 25 : KERNEL SOFTWARE INTERFACE

Notebooks may be exported to a range of static formats, including HTML (for example, for blog posts), reStructuredText, LaTeX, PDF, and slide shows, via the [nbconvert](https://nbconvert.readthedocs.io/en/latest/) command.

Furthermore, any .ipynd notebook document available from a public URL can be shared via the [Jupyter Notebook Viewer](https://jupyter-notebook.readthedocs.io/en/latest/nbviewer) ([nbviewer](https://nbviewer.jupyter.org/)). This service loads the notebook document from the URL and renders it as a static web page. The results may thus be shared with a colleague, or as a public blog post, without other users needing to install the Jupyter notebook themselves. In effect, [nbviewer](https://nbviewer.jupyter.org/) is simply [nbconvert](https://nbconvert.readthedocs.io/en/latest/) as a web service, so you can do your own static conversions with nbconvert, without relying on nbviewer.

The overall architecture of the jupyter notebook is as follows



FIGURE 26 : CLIENT SERVER BEHAVIOUR OF SOFTWARE

# 

# 

# 

4. OTHER NON FUNCTIONAL REQUIRMENTS

# PERFORMANCE REQUIREMENTS:

# The Naïve bayes algorithmwill have very less over heads when the targeted categories are very discrete. The continuous nature of the training data can lead to the errors, so the regression algorithms are better alternatives when the data to be trained is complex. The performance will also be affected by the independent constraints in the relation of the attributes to be tested. In the scenario of dynamic decision making, often required in the case of web browsers, sensex estimation etc . this model is quite handy.

### Scalability Requirements:

Scalable refers the ability to upgrade the present by either integrating with

bigger things or by increasing the complexity of the stand alone project

in the context of our project we can use it to increase the business value of

project.

### 

### Training requirements:

The people have to be aware of the python libraries.

They have to know regarding the jupytor notebook,how to operate it

They have to be able to conceptualise and analyse the algorithm in the

context of the project scope

# 

### 5. REFERENCES

* 1. Text classification Using machine learning techniques by WSEAS
  2. Big Data Imperative by Soumendra Mohanty, Madhu Jagadeesh, and Harsha Srivasta.
  3. Artificial Intelligence A Systems Approach by M. Tim Jones. <https://www.analyticvidhya.com/blog/2017/09/naive-bayes-explained>.
  4. https://jupyter-notebook.readthedocs.io/en/latestlnotebook.html.
  5. https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what\_is\_jupyter.html.