A Hybrid Approach for Classification and Feature Extraction Using Machine Learning Techniques

Asim Alvi

01-241182-057

Master of Science (Software Engineering)

Department of Software Engineering

Bahria University Islamabad, Pakistan

2021

A Hybrid Approach for Classification and Feature Extraction Using Machine learning Techniques

Submitted by:

Asim Alvi

01-241182-057

Supervised by:

Dr. Kashif Sultan

Co- Supervised by:

Dr. Adeel M. Syed

Master of Science (Software Engineering)

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Science (Software Engineering) at Bahria University Islamabad

Department of Software Engineering Bahria University

Islamabad, Pakistan 2021

Plagiarism Undertaking

I take full responsibility for the research work conducted during the MS thesis titled **"A Hybrid Approach for Classification and Feature Extraction Using Machine learning Techniques"**. I solemnly declare that the research work presented in the thesis is done solely by me with no significant help from any other person; however, small help wherever taken is duly acknowledged. I have also written the complete thesis by myself. Moreover, I have not presented this thesis (or substantially similar research work) or any part of the thesis previously to any other degree-awarding institution within Pakistan or abroad.

I understand that the management of the Department of Software Engineering has a zero-tolerance policy towards plagiarism. Therefore, I, as an author of the above-mentioned thesis, I solemnly declare that no portion of my thesis has been plagiarized and any material used in the thesis from other sources is properly referenced. Furthermore, the work presented in the thesis is my own original work except dataset and I have positively cited, the related work of the other researchers by clearly differentiating my work from their relevant work.

I further understand that if I'm found guilty of any form of plagiarism in my thesis work even after my graduation, the University reserves the right to revoke my MS degree. Moreover, the University will also have the right to publish my name on its website that keeps a record of the students who plagiarized in their thesis work.

_____

Asim Alvi

Date: _____

Certificate of Originality

I, **Asim Alvi**, hereby state that my Master's thesis titled **"A Hybrid Approach for Classification and Feature Extraction Using Machine learning Techniques"** are the product of my own research work except, as cited property and accurately in the acknowledgements and references, the material taken from such sources as research journals, books, internet, etc. solely to support, elaborate, compare and extend the earlier work. Further, this work has not been submitted by me previously for any degree, nor it shall be submitted by me in the future for obtaining any degree from this University, or any other university or institution. The incorrectness of this information, if proved at any stage, shall authorities the University to cancel my degree.

Signature: _____ Date: _____

Name of the Research Student: ____Asim Alvi_____

Thesis Completion Certificate

Scholar's Name: Asim Alvi

Registration No.: 01-241182-057

Program of Study: Master of Science (Software Engineering)

Thesis Title:

A Hybrid Approach for classification and feature extraction using Machine learning techniques

---

It is to certify that the above student's thesis has been completed to my satisfaction and, to my belief, its standard is appropriate for submission for evaluation. I have also conducted a plagiarism test of this thesis using HEC prescribed software and found a similarity index at ___12%_____ that is within the permissible limit set by HEC for the MS degree thesis. I have also found the thesis in a format recognized by the Bahria University for the MS thesis.

Principal Supervisor's Signature: _____

Name: _____ Date: _____

Abstract

Artificial intelligence (AI) solutions are used to help make choices that include a high precision of choices they recommend and a deep understanding of choices so that the chiefs can trust them. Verifiable, non-emblematic learning methods have greater perceptual accuracy. Hybrid AI systems analyze the data and exploratory characteristics of approach types. The fundamental purpose of this commitment is to differentiate between a proper AI strategy for choice of assistance that produces reliable and fair outcomes, depending on the various AI techniques, which provide an analysis of different approaches. In this study, we present a hybrid framework for classification and feature extraction. Such a hybrid framework is required for the selection of dataset to Machine learning classifier that will have different results with unlike datasets We have utilized five different types of datasets in this study. As datasets are imbalanced so preprocessing of data is performed firstly. Input with maximum accurate results will be reproduced from our hybrid approach because this approach shows which type of classifiers should be used under what type of dataset, meanwhile exception of the generous fact based on results should be different among different classifiers when applied to a various dataset. After that, a comparative analysis of generic Machine learning algorithms with various datasets has been made as well. The accuracy of the hybrid approaches is compared with the generic approaches, a clear improvement in results in the form of accuracy, precision, recall, and F1 score is found. We used the initial layers of CNN for feature extraction and passess them to ML algorithms for classification.Each information image will go through a progression of convolution layers with channels (Kernels), pooling, fully linked layers (FC) and applying Softmax capabilities to define an object with probabilistic qualities anywhere in the range of 0 and 1. It's one of the simple classes for acknowledging images, arrangements for photos. At the end of this study, we infer which Machine learning algorithm improves the classification accuracy for which type of dataset.

Keywords: Machine learning, Convolution neural network, classification accuracy.

Dedication

This thesis is dedicated to my parents for their love, endless support, and encouragement.

Acknowledgments

First of all I am obliged to Allah Almighty the Merciful, the Beneficent and the source of all Knowledge, for granting me the courage and knowledge to complete this document.

I am sincerely grateful to **Dr. Kashif Sultan and Dr. Adeel M Syed** for providing me an opportunity to Undertake project work in his esteemed company and practical training which will go Long-term in shaping my career in future.

I also express my deepest respect to my teachers especially Dr Kashif Sultan, Dr Adeel M Syed and Dr Ahmad Ali, Dr Tamim Ahmed Khan, Dr Awais Majeed for their guiding support and all faculty for giving a very patient hearing whenever I needed it. Their contribution is significant for the emergence of this project.

I am deeply indebted to all the fellows for their valuable contribution during the academic session and guidance in preparation for this project.Finally, it is the effort of my parents, teachers and esteemed friends.

Asim Alvi

01-241182-057

Table of Contents

LIST OF FIGURES

List of Tables

List of Abbreviations

CNN	**Convolutional Neural Network**
ASR	**Automatic Speech Recognition**
CSV	**Comma Separated Values**
PCA	**Principal Component Analysis**
DL	**Deep Learning**
NLP	**Natural Language Processing**
CNN	**Convolutional Neural Network**

1.1 Chapter 1

1.1 Introduction

Computational effectiveness in displaying and anticipating is a certain degree of leeway over certain other arrangement calculations, which is possible due to the fact of simple parallelization, particularly useful for massive data [1]. With regard to the cited qualities, it is advisable to emphasize two more: security like overfitting and ability to take care of a large given trait without having to select them. The Naive Bayes classifier is defined as computational competence, low fluctuation, constant learning, and direct prediction of back probability, clamour strength and resilience of missing characteristics [2].

Text characterization is a large and fundamental undertaking in controlled Artificial Intelligence (AI). Its application is in the field of email spam recognition, slant inquiry, language area, and character development and so on. Various classifiers can be used to characterize the archive. Many of them are neural networks, support vector machines, genetic equations, Naive Bayes classifiers, k-Nearest Neighbour (KNN) and Rocchio classifier [3].

K-nearest neighbor is a unilabiate calculus of lethargic learning. Since it is a non-parametric computation, it makes no assumptions about the basic dispersion of information. It is a crucial leeway, because much of the down-to - earth knowledge does not comply with the theoretical assumptions made, and this is where non-parametric measurements like the nearest neighbor play the hero. K-nearest neighbor is also a lethargic calculation which indicates that it does not use the emphasis of the planning info to make any inference [4]. Subsequently, the initiation phase is ultimately fast. Lack of uncertainty means that K Nearest Neighbor (KNN) holds all the details on the planning. KNN shall make a choice based on the planning of the information set as a whole.

In order to enhance the classification accuracy of the nearest neighbor, a modified method of recognition of different components is discussed. Highlight choices are difficult choices that should be made in a number of territories, particularly in the case

of man-made reasoning. The key issues in creating high-profile determination processes are the collection of a small list of capabilities to reduce the cost and running time of the framework, as well as the achievement of an acceptably high recognition rate [4].

In recent years, a variety of post-pruning estimates have been provided, such as reduced blunder pruning, error-free pruning, less blunder pruning, and cost-based pruning. For example, most of the culling methodologies decreased the error culling and the least blunder pruning of the selected tree in the base on request evaluating the misclassifying blunders for each over fitting [5].

In general, the planning details will not be explicitly distinct, that is, there will be no insulating hyper plane in the data space. In those kinds of instances, we can use a variation based bit capacity to change settings to a higher dimensional component space, where it is obligated to be directly separate (or potentially reduce the amount of tend to group errors). The expected bit trick encourages us to make such an improvement inside the SVM, despite having to take care of a huge penalty as far as computational skills are concerned [6].

The degree to which a person expresses his or her expectations is limited on a regular basis when individuals need to give assessments on an item in the form of score / star assessments. In any case, whenever an individual is allowed to interact audits in the sort of system interfaces, he can be highly accurate as to what dimensions to the object are appropriate and what is certainly not normal [7].

1.2 Deep Learning

Deep Learning is a subset of Machine learning, which then again is a subset of Artificial Intelligence, where neural network organizations, calculations inspired by the human cerebrum, gain from a lot of information. It encourages a computer to channel contributions through layers to figure out how to foresee and characterize data. Perceptions can be as pictures, text, or sound. The motivation for deep learning is the way that the human cerebrum channels data. Fig 1 shows the process of Deep learning given below**[40]**.

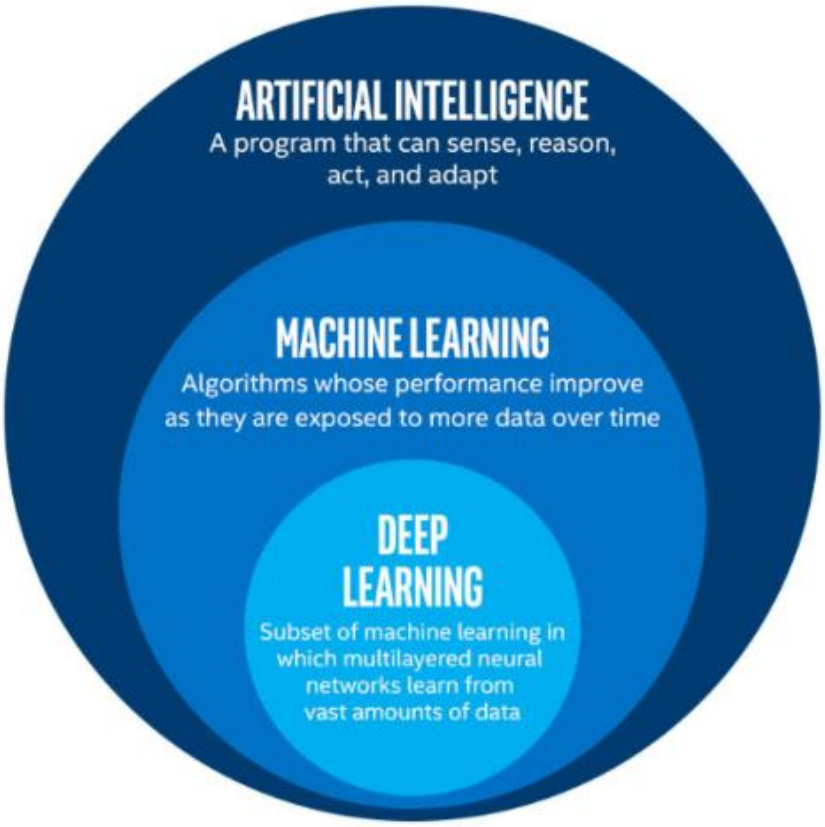


Figure 1 Deep learning:[40]

1.2.1 Convolutional Neural Network

Convolutional neural network (CNN) is a class of deep neural organizations, most normally applied to examining visual symbolism. Convolutional networks were motivated by natural cycles in that the availability design between neurons takes after the association of the creature visual cortex deep learning CNN Approaches to prepare and test, each information picture will go it through a progression of convolution layers with channels (Kernels), Pooling, completely associated layers (FC) and apply Softmax capacity to characterize an item with probabilistic qualities somewhere in the range of 0 and 1. It's one of the fundamental classes to do pictures acknowledgment, pictures arrangements. Items identifications, acknowledgment faces and so forth, are a portion of the zones where CNNs are broadly utilized. In CNNs, the principal layer is consistently a Convolutional layer. These are characterized utilizing the three spatial measurements: length, width, and profundity. These layers are not completely associated – implying that the neurons from one layer don't interface with every single neuron in the accompanying layer. The yield of the last convolution layer is the contribution to the main completely associated layer**[40]**.

1.2.2 Recurrent Neural Network

A RNN is utilized for situations where the information contains fleeting properties, for example, a period arrangement. Additionally, where the information is setting touchy, as on account of sentence finish, the capacity of memory gave by the input circles is basic for satisfactory execution. RNNs are utilized in deep learning and in the improvement of Approaches that recreate the action of neurons in the human brain**[40]**.

1.2.3 Artificial Neural Network

Artificial Neural Network (ANN) is capable of learning any nonlinear function. Thus, these organizations are famously known as Universal Function Approximators. ANNs have the ability to learn loads that map any contribution to the yield. . It works like the manner in which human cerebrum measures data. ANN incorporates countless associated preparing units that cooperate to handle data. They comprise of an information layer, different hidden layers, and a output layer**[40]**

1.3 Machine learning

AI is a form of man-made logic that focuses basically on AI based on experience and makes perceptions contingent on their knowledge. It encourages workstations or machines to decide on information-driven decisions, rather than being specifically configured for the completion of a particular endeavor. These initiatives or computations are designed in such a way that they learn and improve after a certain time with a new set of information. Fig 2 shows the process of Machine learning given below**[40]**.

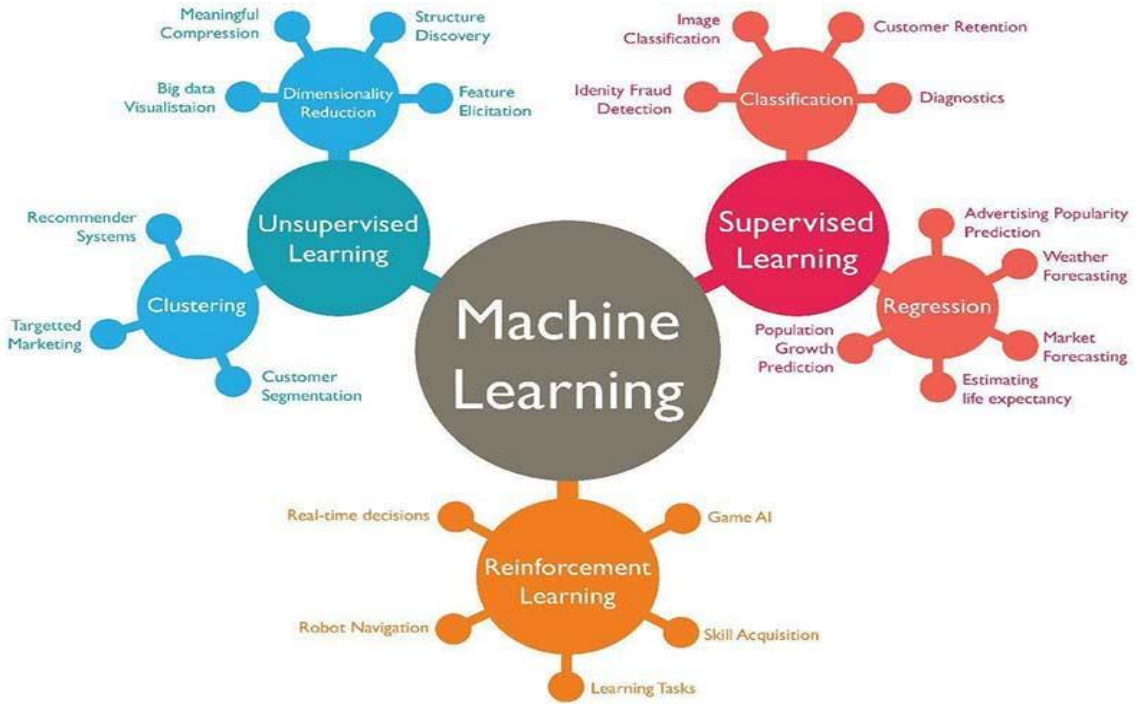


Figure 2 Machine learning

1.3.1 Naive Bayes Classifiers

Naive Bayes Classifiers were used for multi-name revelation, where the preparation of informational collections consists of only several instances if each name is linked from one threshold and another, or the endeavor is to predict an array of names from indiscernible cases. In simple terms, the Naive Bayes classifier expects the proximity of a particular item in a class to be separated. The proximity of a different component. Irrespective of whether high points rely on each other or on the existence of varying highlights, these properties automatically add to the probability. The naive Bayes Approach is far from complicated to manufacture and particularly useful for unusually big amounts of data and information. In addition to simplicity, Naive Bayes is recognized to outsmart sometimes. Fig 3 shows the working of Naïve Bayes given below**[40]**.

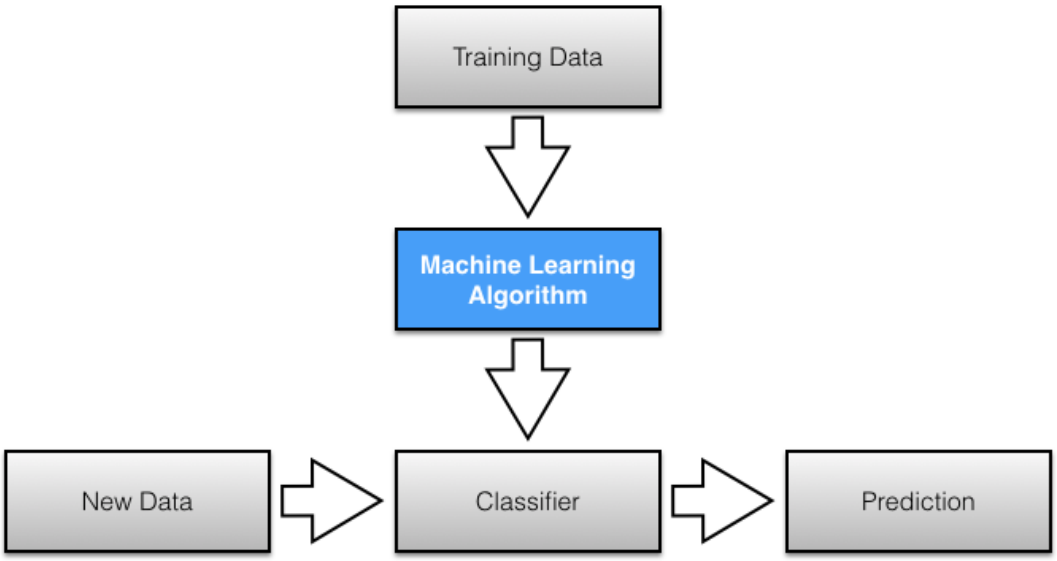


Figure 3 Naïve Bayes Working:[40]

1.3.2 Logistic Regression

It is referred to as the capacity used at the heart of the strategy, the strategic capacity. The measured capacity, furthermore called sigmoid capacity, was generated by researchers to represent the characteristics of population growth in nature, increasing rapidly and optimizing at the earth's forth that. It's an S-shaped twist that can accept any real valued quantity and direct it Opportunity also in the range of 0 and 1, but never exactly at such financial limit.

$$\frac{1}{1 + e^{-value}}$$

In which e is the basis of the character trait logarithms (Euler's number or EXP) (work in your spreadsheet) and is worth the true numerical value that you need to change. The belows is a plot of the figures between-5 and 5 modified to range 0 and 1 using the measured power. Logistic regression graphs shows in fig 4 given below**[40]**.

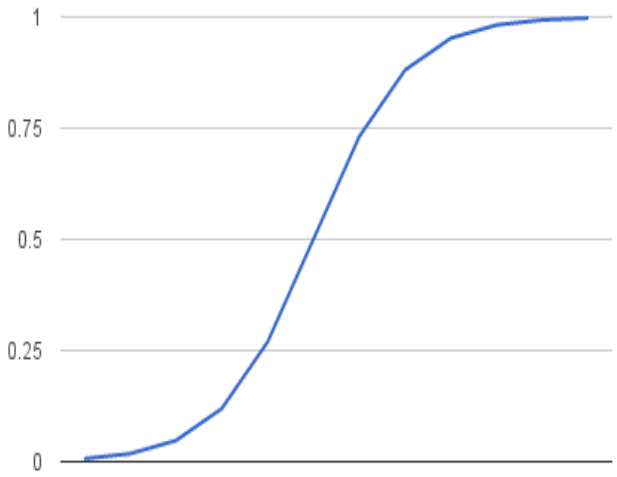


Figure 4 Logistic Regression Graph:[40]

1.3.3 Decision Trees

Decision trees create classification or reversal Approaches as a tree - like structure. This divides the knowledge index into smaller and smaller subsets, while at the same time the associated option tree is continuously evolving. The final result is a tree with selected nodes and leaf nodes. A selection node needs to have at least two parts, and a leaves node shall talk to a classification or a preference. The greatest-choice decision is the tree that corresponds to the primary judge called the root node. Choice trees can handle both unalloyed and matrices information. Fig 5 shows the working of the Decision tree given below**[40]**.

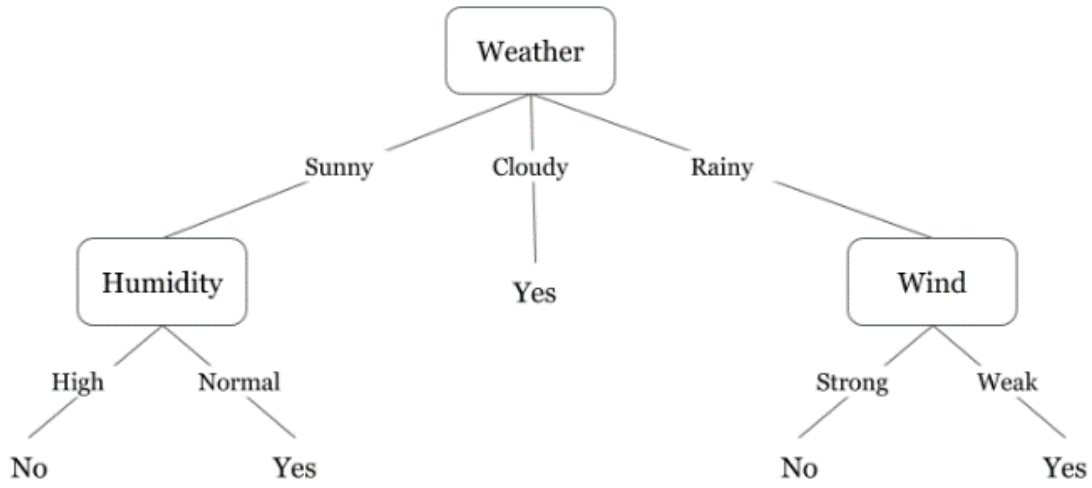


Figure 5 Decision tree working:[40]

1.3.4 Support Vector Machine

In AI, support vector machines (SVMs, thus support vector systems) are controlled learning Approaches with associated learning measurements that decompose additional data used for order and rebound study. Support vector Machine Computation is feasible for modified classification, despite not performing well enough on imbalanced datasets. The SVM measurement finds a limit of choice for the feature space that easiest parts the systems into 2 classifications. The split is rendered sensitive using an edge which allows for misclassification of a few targets. Fig 6 shows the working of SVM given below**[40]**.

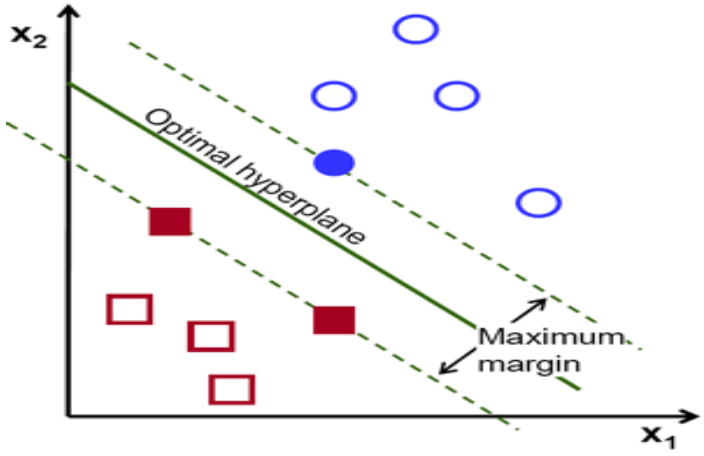


Figure 6 Svm working:[40]

1.3.4 Random Forest

Random forest or random choice of algorithm is a collectivist learning technique for order, recurrence and various assignments, which function by building a huge number of choice trees in the preparation of time and producing the class which is the system of classes (description) or mean expectation (relapse) of the individual trees. Arbitrary option of backwoods to match the propensity of preference trees. Random forest is measurement of the AI outfit. Because of its excellent or impressive appearance over a diverse variety of order and rebound perceptive showing problems, it is probably a very well-known and generally used AI calculation. It is also simple to use, as it has hardly any primary parameters and rational methodologies to organize. Fig 7 shows the working of Random forest given below**[40]**.

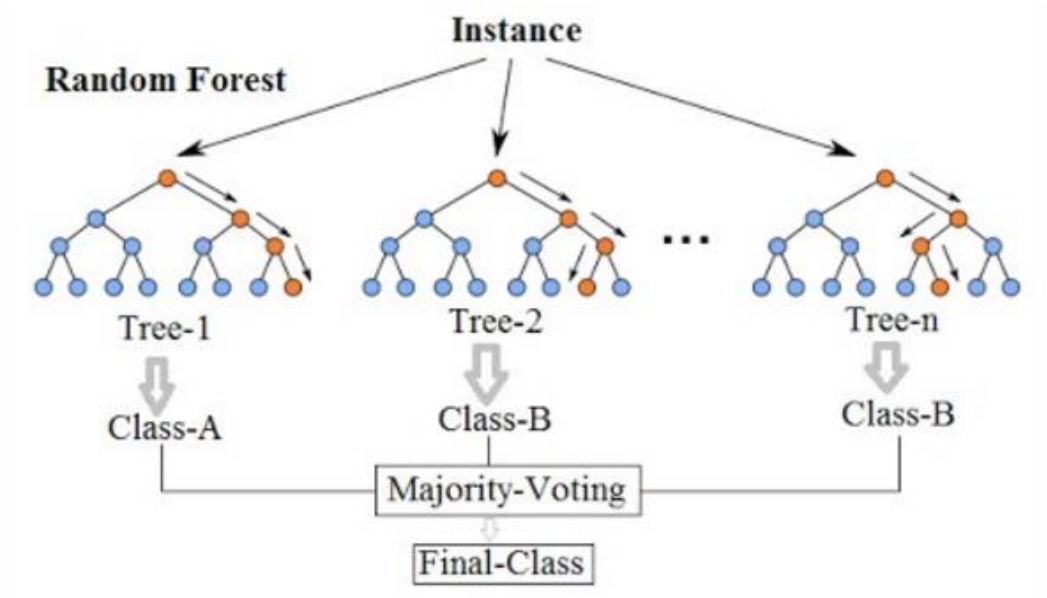


Figure 7 Random forest working:[40]

1.4 Datasets

We have used the following datasets in our research. These datasets were publically available.

- Dataset 1 : Dataset consists of Amazon mobile reviews.
- Dataset 2 : Dataset consists of Breast Cancer Records.
- Dataset 3 : Dataset consists of Amazon Alexa Product reviews.
- Dataset 4 : Dataset consists of bi-language tweets English + Roman Hindi.
- Dataset 5 : Dataset consists of drugs records used by patients during the specific symptoms.

These datasets are explained in detail belows, one by one.

1.4.1 Amazon Mobile Reviews Dataset

Extracted 400 thousand reviews of unlocked mobile phones sold on Amazon.com to find out insights with respect to reviews, ratings, price and their relationships.
given belows are the fields:

- Product Title
- Brand
- Price
- Rating
- Review text
- Number of people who found the review helpful

Cell phones have upset the manner in which we buy items on the internet, making all the data accessible readily available. As the admittance to data gets simpler, an ever-increasing number of purchasers will look for item data from different customers separated from the data given by the merchant. Surveys and appraisals presented by shoppers are instances of such sort of data and they have just become a necessary piece of the client's purchasing choice cycle. The audit and evaluations stage given by eCommerce players makes a straightforward framework for customers to make educated choices and feel certain about it.

1.4.2 Breast Cancer Dataset

Around the world, bosom disease is the most widely recognized sort of malignant growth in ladies and the second most noteworthy regarding mortality rates. Diagnosis of bosom malignant growth is performed when an unusual protuberance is found (from self-assessment or x-beam) or a small bit of calcium is seen (on a x-beam). After a dubious protuberance is discovered, the specialist will lead a conclusion to decide if it is harmful and, assuming this is the case, regardless of whether it has spread to different pieces of the body.
Loading Breast dataset from the Scikit-learn “load_breast_cancer” Class
This bosom malignant growth dataset was acquired from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg.s

1.4.3 Amazon Product Reviews Dataset

This is a rundown of more than 34,000 purchaser surveys for Amazon items like the Kindle, Fire TV Stick, and more given by Datafiniti's Product Database. The dataset incorporates fundamental item data, rating, audit text, and more for every item.

1.4.4 Drug Analysis Dataset

The main aim of the dataset is to predict the medicine based on the condition of the patient on the basis of previous records.
161297,7 Total Number of shapes of train records, Shape of Test 53766,7
Unique id, drug name, condition, review, rating, date, useful count

1.4.5 Bilanguage Dataset

The primary point of the task is to build up an assessment analyzer that can be utilized on twitter information to group it as sure or negative. Our undertaking deals with the test of bilingual remarks, where individuals tweet in two dialects, for this situation Hindi and English, in the Latin Alphabet.
This task examines the various classifiers that can be utilized for supposition examination of twitter information, to order the tweets as sure or negative. The test of Hindi-English Code-blended Social Media Text is centered around here. A current marked Kaggle dataset is utilized for this investigation. 40,000 columns of this dataset are haphazardly chosen and afterward cleaned. Descriptive words, intensifiers and conceptual things are chosen as highlights and removed for each cleaned tweet.
The capacity likewise attempts to deal with the test of language uncertainty which happens when a word exists in both the English and Hindi word references. The deciphered last string is passed to another capacity that utilizes the string as an experiment for the seven base classifiers and the half and half approach and the conclusion anticipated by every one of these classifiers is printed alongside the extricated highlights and the interpreted last tweet/string given by the client.

The following table presents a summary of these datasets and provides the Train to Test ration that we have used for them. Table 1 shows the Summary of all datasets given belows.

Table I Summary of datasets used

Sr. No.	Dataset	Details of Dataset	Train/Test Ratio
1	Amazon Mobile Reviews	400,000 Records	70/30
2	Breast Cancer Dataset	56,000 Records	70/30
3	Amazon Alexa Products	34,000 Records	70/30
4	Bi-language Dataset	40,000 Records	70/30
5	Drugs Record Dataset	161297 Records	70/30

1.5 Motivation

Early application of deep learning will save some time for the data filtering from the raw data. To achieve excellence in the classification and features extraction of different datasets by transforming the traditional methodologies into a more effective and comprehensive solution which will ultimately bring a breakthrough in lab sciences in a cost-effective and more accurate manner. And we cover the existing techniques issues, for the raw data set of different domains of labs have sharp light, and do comparative analysis by enhancing the previous techniques and also by comparing with our hybrid Approach . So that a clear experimental research will outcome.

1.6 Problem statement

To use a hybrid approach using Machine Learning (ML) and Deep Learning (DL) for improvement in accuracy and training time as well as decrease in error rates in terms of misclassification of data.

1.7 Aims and Objective

Our aim is towards making a hybrid approach using different ML and DL algorithms in order to compare them with some preprocessing techniques on immature or raw dataset. We also wanted to increase the accuracy of classification and minimize the misclassification errors in the results. Lastly we wanted to perform a comparative analysis on which classifier is working better on which dataset.

The objective of this thesis/Research is:-

To Propose a comparative analysis of different implementations of Machine learning and deep learning algorithms through Hybrid Approach on different datasets with data normalization of raw dataset with some hybrid techniques.

1.8 Research Questions

For the above-mentioned proposed solution following questions will be addressed.

1. How to select a classification algorithm to obtain better results against a particular dataset?
2. Which Machine learning technique will improve the classification performance if the details of a dataset is unknown?
3. How will a hybrid framework be helpful in reducing the classification error?

1.9 Contributions

1. Selection of Machine learning classifiers for better results of classification against dataset.
2. Shows which approach and technique will improve the results if you don't know the details regarding the dataset.
3. Hybrid Approach working against different datasets to see the accuracy and misclassification rate also the comparative analysis of results in terms of precision, recall, accuracy and f1 score.
4. Shows a significant impact of data filtering techniques against a classification problem.
5. Hybrid approach using deep learning and Machine learning techniques for better accuracy.

1.10 Document Structure

The remaining document is break down as follows:

Chapter 2 Related Work: in this chapter we will go through the literature that shows how much work has already been done in this area of Machine learning. We had discussed the different techniques used by different researchers in different areas and how these techniques were significantly involved in Machine learning.

Chapter 3 Methodology goes through the methodology used in taking this research work. This chapter describes the outcome of the experimental planning phase, including data collection procedures, data filtering, data analysis procedure and evaluation of the validity through cross validation using metrics.

Chapter 4 Design and Experiment describes each step in the production of the experiment, including the sample, preparation of algorithms, techniques, data collection performed with mapping to the input requirement and validity procedure.

Chapter 5 Analysis in this chapter we have discussed and thoroughly analysed the results we got from the experiment and deducted whether our technique of comparison of different Approaches work and which one shows best results according to the dataset or input either this is effective in obtaining the best output or not.

Chapter 6 Conclusion and Remarks: This chapter takes the discoveries after the experimental analysis including an assessment of results and interpretations, limitations related to the study, inferences and lessons learned.

Chapter 1

Literature Review

2.1 Introduction

This chapter includes the history related to Machine learning, computer supported techniques for prediction, and Learning based classification and past work. In this chapter we will go through the literature and find out the importance of Machine learning techniques and different classification algorithms based on different datasets and their applications in the computing field. Different research scientists had proposed different techniques for Machine learning algorithms for different areas either Medical, IOT, or much more. Recent predictions are made for the Covid-19 No of users that may be affected grounded on the data science and Machine learning a techniques using for forecast of affected people and to obtain the finest solution for grouping the different people either affected or not, we have tried to go through all the techniques proposed in the literature and touch the boundary of existing domain to our best knowledge.

2.2 Background

There exist several types of malignant growth on the planet, unique of which is colon disease. Colon's malignant development is single one of the major life-threatening diseases on the planet. In either case, patients often struggle to recognize this disease on the grounds that there are no visible side effects of colon malignancy in the early stages, and individuals do not believe they are experiencing it. Colon disease is one of the most well-known dangerous tumors with a high frequency in the age gathering of 40-50 years and is an intense danger to human life and wellbeing. In 2017, around 136,000 new instances of colon malignant growth were analyzed. Around 1 out of 20 individuals will create colon malignant growth during their lifetime [8].

Since the dataset takes 127 Metallica and 80 Nirvana tunes, the Naive Bayes Classifier has been castoff on the grounds that it is reasonable to use a small dataset [9].

The Naive Bayes Classifier of ICA can capture and effectively label the innate characteristics of the final images. The remarkable characterization error rate of the tests indicates that the use of naive Bayesian Classifiers is acceptable for surface classification. As a result, the proposed equation can be organized as solid, modifying each of the 28 images would generate new concept capacities. It managed to bring about the capacity of the premises. The development of the ICA premises approaches is conducted following the application of a large-scale reduction of observations as provided by the PCA using the Naïve Bayes Classifier [10].

The calculation was designed to predict repeated opportunities for bosom disease using the Wisconsin prognostic dataset. The developer used the Guileless Bayes classifier to describe irregular instances. A new estimate using an innocent Bayes seeing the on bosom disease expectations has also been developed. The author used details on breast cancer screening for the predictions of the tumor. GUI was designed to discover the possibility of malignant growth in the bosom in women later on. They used a total of nine attributes for prediction. Another parameter choice technique was created by bundling and ordering. In this research, the author focuses with approach-based learning, as well as the authors compare and other time - varying-choice approaches on the underway dataset [11].

Bokde et al. have transmitted the R code for showcase. Due to the roughly similar thought experiment making plans of the PSF, the measurement of the k-nearest neighbor (KNN) is an elderly thought experiment Machine And is flawlessly updated [12]. It is often used to deal with non-linear problems, such as Credit score and bank customer rankings, in which the information collected does not necessarily obey the conventional straight assumption, and it should be one of the first decisions when there is almost zero earlier knowledge on distribution information. In addition, the impact of the variables on the test types can be effectively minimized [13].

KNN has sophisticated predicting accuracy and takes no conventions for the composed data, and mainly, it is not penetrating to the outliers. It has been extensively practical in actual-ecosphere glitches, like investigating the area lying under the stock market, so that daily up and down of shares in the market [14].

KNN-grounded classification accuracy might expressively be contingent on the rules that are executed to compute reserves between training and testing samples [15]. Towards discourse on this subject, the biased reserve rules and the depiction-based approaches are used to calculate the resemblances of samples to regulate k-nearest neighbours of apiece demanded mock-up [16].

KNN strategy is such a well-known information extraction and insight classification technique due to its easy application and impressive arrangement execution. It is, however, irrational for traditional KNN approaches to demote fixed K confidence (regardless of the fact that it has been developed by experts) to all tests [17].

Decision Tree learning customs a classification tree in place of an insightful technique that plots expectations of a thing to judgments about the value of an object. It is one of the perceptive showcase methods used in measurement, information mining, and AI. Tree structures where the subjective parameter may take the shape of a given configuration of attributes are called order trees, In this tree structure, the leaf relate to the identities of the class and the nodes speak to the conjunctures of the highlights which lead to those class marks which are opposed to the ones used to choose the yield for the classification [18].

In Breiman 's technique, the tree in the assortment is formed by choosing a small selection of information (additionally referred to as highlights or factors in the future) to be part of and, respectively, by selecting the best part that depends on these landmarks in the preparation package. The tree is formed using the most severe CART technique, without trimming. This dimensionality random sampling plot is mixed with sacking to reassemble, with partial replacement, the preparation of the instance shows almost every time another tree is formed [19].

In [20] Jung and Yoon carried out an analysis of the members in particularly unique motels and ensured that the satisfaction of the place was associated with the intention of retention. He looked at how coordinating a mistake Executive Culture (EMC) could lead to higher job performance and lower retention expectations. Labrague et al. studied the retention of nurses in the Philippines and observed that age, Work performance and

occupational stress have been the most significant elements in the anticipation of retention.

In [21] the framework for the analysis of different methods, i.e. the random forests, the exceptional learning machine, the convolutional neural network and the boost vector machine, is implemented in an attempt to discover the most capable one. Random forests have been shown to surpass similar classification approaches in terms of accuracy, steadiness and warmth of recognition, particularly with a little sequence of planning and preparation. As well, contrasting and traditional approaches, random forest are not easily influenced by ecological clamour.

The biggest problem with re-sampling techniques is that, on the basis of few cases, more knowledge analysing or breaking up will lead to substantial progress. To catch up with this extra vulnerability, Isaksson et al. inform the use of certainty span, guaranteeing that the re-sampling of evaluations is unreliable. In either case, Kohavi found both the bootstrap and the cross-approval of decently calculated standard knowledge collections and suggested 10-fold cross-approval as an acceptable accuracy estimation method [22].

The least complicated method comprises having to run RF with default boundary values, since no combined and straightforward to-use tuning method has yet been established. It is not the aim of this paper to talk about how to enhance RF presentation by fitting tuning techniques and what level of skill is indisputably required to use RF [23].

SVM is a Machine learning classification technique which implements the supervised approach for learning and it is widely used for cancer diagnosis and prognosis field. The SVM technique isolates two classes by choosing a normal classifier that expands the edge. This partition is referred to as the ideal hyper plane insulation. Regularization demarcation and bit work are the two important sections which need to be addressed before guiding the preparation process. A portion of the critical investigations used by SVM to detect the bosom disease used Heuristic analysis SVM approaches, for instance, smooth SVM, wide SVM and general highly nonlinear SVM [24].

Sentiment analysis investigation engines parse through these literary survey data and generate yields in the type of polarity, e.g. positive, negative or neutral. These aids in the determination of the purposes of the important variances in the dealings of the goods and they can be rectified accordingly. Computations in orders may have an impact on the accuracy of the direct consequence in the extreme point and, consequently, on the insufficiency of the grouping [7].

In the incident that accuracy is most noticeable to us, at that moment we should be leaning towards a classification approach similar to the Random Forest that manipulates strong learning time nevertheless has the finest accuracy. Mostly on the off chance that the preparation of control and cognition is a matter, the Naïve Bayes classifier would be nominated at that moment for its low memory and the need for power handling. On the off possibility that little time to prepare is available but you have the wonderful managing structure and memory then peak Entropy tends to end up being a commendable alternative [25].

This achievement develops more blunder-free packed aside from DT by encouraging the normalization process by ejecting weak apprentices and accelerating Hybrid-target work, the discoveries may not be directly appropriate for different types of AI calculations in anticipation and character development as the unique half-and - half-approach may contain various credits for evaluation; In addition, the information arrangement was directed to consider 13 boundaries, such as info and four calm, tropical, cold and hot-bone-dry atmospheres. This calls for further review by endorsing the approach for different settings and using larger examples covering numerous structural vitality limits and atmospheres [25].

The fundamental work of this paper shows the order of data for emotional scriptures, that is twitter and item appraisals. The data is separated and hooked on as positive and negative. Techniques like Support Vector Machines (SVM), Conditional Random Field (CRF) and Naive Bayes Multinomial (NBM) remain the calculation prediction based on supervised learning. Three classifiers are individually designed by separating specific capabilities on basis of features [26].

The SVM classifier, prepared on an unprovoked dataset, will build flawed approaches that are yet another-sided against the large class and have low execution on the smaller class, as a large part of all the other classification approaches [27].

A clear multi-name classification technique is dual significance (BR) approach, which breaks the issue down into a series of singular-name multiclass classification issues. A lot of multi-class classifiers are described along these lines which were used to make assumptions. This simple technique, however it may be, absolutely rejects the conditions between different names. Virtually speaking, multiple publications in a pattern, (For instance, a picture) can have solid relationships or requirements. For example, if a ship classification is shown in a photo, the water class is almost certainly the same in that picture. Misappropriating such trademark dependency could significantly improve the implementation of standards for a multi-name classification [28].

In [29] by evaluating the efficiency of different popular sentiment classification methods as well as developing a comprehensive method that further enhances sentiment classification accuracy. Slight effort has been done in the arena of twitter sentiment analysis of air carriers. This previous work contrasts a variety of different conventional classification techniques and chooses the most specific approchling approach for the implementation of sentiment classification. However, the whole approach that we express increases performance by integrating these sentiment classifiers. In the field of air carriers, the precision of the sentiment classification is high enough just to investigate customer satisfaction. This approach refers to the study of Twitter data by air carriers about their quality of services.

2.3 Critical Analysis

Table 2 shows the Previous work of different authors given belows.

Table 2 Previous work critical analysis

Year	Authors	Contribution based on classifier and dataset with respect to attributes	Algorithm / Classifier/ Approach/ Techniques	Used Dataset	Metrics Used
2015	M. Shahid et al. [13]	Used 3 classification Approachs: naïve Bayes, Decision tree and KNN. Naïve Bayes achieved finest in terms of higher accurateness, higher precision, higher recall and higher recall and higher value of F-measure as compare to the decision tree and KNN.	Naïve Bayes, Decision tree and KNN are the techniques used.	Yes	Precision, Recall, F1
2020	H.Ahmad et al.	Using lexicon Multinomial KNN and SVM algorithms are used for this purpose. Both supervised and unsupervised approach is used. SVM performs the highest accuracy of 82% for validation and KNN performs low accuracy of 26%.	KNN and SVM.	Yes	Accuracy
2020	G.Yasa et al. [15]	Performed sentiment analysis on social media product reviews. Use two classification approaches for calculating accuracy one is Naïve Bayes and another one is Multinomial naïve Bayes. Naïve Bayes performs the best accuracy on the given dataset.	Naïve Bayes and Multinomial Naïve Bayes.	Yes	Accuracy
2019	K.Mehmood	Used machinee learning algorithms with a different type of word-level feature, character level features and combination of both. And also performed CNN and LSTM on the whole dataset.		No	Performance

2018	R.Ankita et al. [17]	Used US airline data set. Used 7 diverse classifiers such as Decision tree , SVM , K-nearest neighbors, Gaussian naïve Bayes, Random Forest, and Logistic regression. Random Forest performs high accuracy which is 85.7%, so random forest is good for this type of data set.	Decision Tree, SVM, KNN, Gaussian Naïve Bayes, Random Forest, and Logistic Regression.	Yes	Accuracy
2018	k. Rida et al. [2]	Data collected from twitter for the airline business, gathered the airline tweets from 4 regions: India, America Europe, Australia, different countries. Three classifiers are used for classification of the dataset, which are Random forest, logistic regression and decision tree. Maximum classification accurateness of 99% observed in the random forest on the given dataset.	Random Forest, Logistic Regression and Decision Tree.	Yes	Accuracy
2015	Isah et al. [3]	Used dataset of drug and cosmetic products. Uses two approaches Lexicon base method and Naïve Bayes. They just do a comparative study about the lexicon base method and naïve Bayes. In the end, they could not come to a conclusion which method is the best	Naïve Bayes.	Yes	Comparison
2015	Eman-M.G. Younis	Microblogs data collected from online reviews about two retail stores in the UK name Tesco and Asda store over Christmas dated 2014. Used The Lexicon base method that is an open-source approach, through which Micro flex data from Twitter is collected, Pre-processed, analyzed and visualized.		Yes	N/A
2013	B.rabia	Used dataset of tweets. Sentiment based filtering applied for specific types using seed lists to reduce the loss of data. Missing data classified through the classifier.		Yes	N/A

2017	Jianqiang et al. [6]	used 5 different twitter data set. Discussed in details the effect of the pre-processing method on classification, and use 6 pre-processing methods using 2 feature Approachs. Used 4 classifiers naïve Bayes, random forest classifier, logistic regression and support vector machinee. Random forest and Naïve Bayes are more sensitive than logistic regression and support vector machinee classifiers for specific data set.	Naïve Bayes, Random Forest classifier, Logistic Regression and Support Vector Machinee.	Yes	N/A
2017	Samonte et al. [7]	Used the twitter data set of Philippine airline services. Use three classifies which are random forest, support vector machinee and naive Bayes. Conduct three experiments, Random Forest continued to provide great accuracy and kappa scores. SVM gives the opposite result to the random forest, SVM gives low accuracy and kappa scores.	random forest, support vector machinee and naïve Bayes	no	N/A
2019	Prabhakar et al. [8]	Use the data set for an airline. Use six different classifiers that are: SVM, Decision tree, Random Forest, Bagging, Boosting, and New Approach. The new approach gives high precision, recall and FI score, according to author's this Approach will not work for other languages.	SVM, aDecision tree, Random Forest, Bagging, Boosting,	Yes	Precision, Recall,
2019	Noor et al. [9]	Data set to use in the locally texted formatted based on roman Urdu language of an e-commerce website draza.pk, which consists of 20.286k reviews. Data set is Split in to 2 sets: 80% of training and 20% of test data to validate. Use SVM (support vector machinee) classifier for the classification of data. Different kernels of SVM are used, cubic kernel archives highest accuracy on the given dataset.	SVM, SVM kernel	Yes	Accuracy
2017	A.Ehsan et al. [10]	Discover the impression of the NLP tool. Different sentimentality features and sentiment lexicon group methods to sentiment polarization classification used. Use 2 different methods compare and a new method (PSWN) proposed for generating a person wordnet.		No	N/A
2018	Geetha	Just overview of the different classifiers. Impact of the classifier on sentiment analysis, tell how classifier used.		No	N/A

2014	Duwairi et al.	Applied 3 classifiers on a dataset, and compare the result which one is best for a specific dataset. The classifier used is SVM, KNN and naïve Bayes. The best precision shows by SVM and the best recall done by KNN.	SVM ,KNN, Naïve Bayes	Yes	Precision,

Man-made brainpower (AI) collaborator has now accomplished impressive milestones in a variety of fields. Essentially, enormous information advancement is the illustrative phase of AI that has shown outrageous accuracy in terms of language recognition, image identification, identification, normal language preparation (NLP) and improvement. Despite the fact that there are several imaginative discovery effects, like craftsmanship, writing and tuning that can't be solved unless it has been turned into human or AI methods. In the field of applications, attempts to illuminate inconveniences that have not yet been able to be grasped or dynamic problems have begun to increase the overall rate of use of Automation. Be that as it may, there may be a lack of record sets to apply gadget acing to the system, and it's hard to know how to unravel the organizing issue. Up to this level. There've been a lot of attempts to see how to organize the computer to collect information about, but there are hardly any inquiries to create a fundamental repository. In this paper, we present the critical device gadget regarding development and suggest a technique for generating network structure records without problems. In addition, the impacts of the programmed time of the defined realities and the impacts of consideration and induction from the comparison dataset are provided in the light of the innovation framework proposed in this work. We took care of the significance of the critical system Machineto analyse methods and document operations that can easily produce insights with the intention of allowing specialists to concentrate on the computer that is familiar with the measurements Nevertheless, to date, device has become more common to citizens in restricted areas, and there are still a range of problems to be addressed. All things considered, there may be a reason behind why AI is turning into the core of the fourth mechanical upset with the overall trend. It's because of the impact of the application of the AI. At present, we are hoping to apply Machineacing to an early stage in the system field, but in the near future, we can be equipped to cure perplexing difficulties faster and more precisely by getting acquainted with gadgets. They accept on the off chance that you'll have to

identify the different system framework Issues with mastering the use of the proposed information generator [30].

Intrusion detection is a fundamental part of security tools, inclusive of adaptive protection appliances, intrusion detection systems, intrusion prevention systems, and firewalls. Various intrusion detection strategies are used; however, their performance is an issue. Intrusion detection overall performance relies upon accuracy, which needs to enhance to lower false alarms and to boom the detection rate. To solve concerns on performance, multilayer perceptron, help vector system (SVM), and different strategies have been used in latest work. Such strategies indicate limitations and are not green to be used in huge information sets, which includes system and network statistics the revelation interruption Machineis used to break down huge traffic information; therefore, a green order method is necessary to defeat the problem. In this paper this difficulty is taken into account. Noteworthy AI algorithms are employed, in particular SVM, irregular forests and an exceptional learning gadget (ELM). Owing to their utility in organization these devices are noteworthy. Because of the fact our day-to-day activities rely heavily on them, interruption detection and preparation are necessary to implement day-to-day and future processes and record structures. Furthermore, future difficulties will prove all the more overwhelming due to the Internet of Things. In this respect, the mechanisms of the location of interruptions have been critical in the past years. A few methods had been used in the identifying structures of interruptions, but in format prescribed, AI methodologies are not surprising. Also, exceptional gadget learning methods have been used, however, some procedures are gradually fitting to break down huge measurements for systems and network frameworks intrusion identification. Unique for handling this issue Specific contraction learning strategies, SVM, RF, and ELM will be discussed and considered in this study ELM discusses different techniques of precision , accuracy, and feedback on the full reality assessments that combine 65,535 observations into activities that include traditional and interfering activities. In addition, the SVM proved favoured effects in half of the aspects tests and in quarter of the data tests over different datasets. In this way ELM is a good technique for the discovery structure of interruptions. That can be aimed at examining an enormous quantity of measures. In future, ELM would be further evaluated to analyse its exhibition in highlighting decision-making and capability-change techniques [31].

This paper leads order measurement according to the client using a pair of disconnected open record extraction and knowledge disclosure tools including WEKA, Rapid Digger, Tanagra, Orange, and Knime. The particular classification of rules, such as Decision Tree, Decision Stump, K-Nearest Neighbor and Naïve Bayes, has also been evaluated for the use of each of the fifth Indian Liver Patient set of data is being used to evaluate the computation of classification in order to collective groups of people both with and without problem of Liver. In this paper we used sets of ILPD (Indian Liver Patient) data set for liver medical data. This has 583 instances of 10 equal variables and one element for analysis of the historical. The overall implementation of the classification techniques on the accuracy evidence was as associated, Accuracy = (TP+TN)/(TP+FP+TN+FN) it is very clear that perhaps the WEKA system estimates the most diminished accuracy for Naive Bayes, but for a similar agreement of rules Knime gadget calculates better accuracy as compared to WEKA. When the Decision Tree Analysis would occur [32].

Since the dispatch of each 1972 of a Landsat-1 primary land explanatory satellite, numerous devices examining calculations had been used to order pixels in the imagery of the Thematic Mapper (TM). Characterisation practices array of parametrically regulated grouping calculations comprising the most extreme probability, unassisted calculations together with ISODAT and k-strategy bundling to device Collecting detailed calculations in neural engineering, decision trees, supporting vector machine and classifying outfits. Different calculations of the type of outfit have been proposed as from late. Random Forest is the most commonly used set of rules for Outfit form. The Random Forest classifier uses bootstrap conglomeration as tree classifiers to structure a class troupe and acceptance tree. Random Forest has been used by a few scientists for land cowl exams. Nonetheless, the Random Forest limit no longer exists. Fully investigate over remote network detection. In this paper we evaluate Random Forest's class accuracy with the other constantly used calculations that incorporate the most probability, least separation, decision tree, neural system, and help classification vector contraction. We propelled reproduction for Random Forest in this investigation and fell to pieces in two Landsat scenes Landsat-eight OLI. The pictures were analysed using ERDAS Imagine, and Liaw and Wiener's guide is managed by the R bundle. From the outcomes it might be evident that Random Forest's general implementation converted into conventional accuracy and kappa co - efficient gestures outstanding to any other classifier. Results indicate that Random Forest was beaten with the neural

system guide and help with Machinevectors. This may have to be a result of contaminated embedded systems. Random Forest works flawlessly despite vast homogeneous records of tutoring, and is reasonably close to exceptions. As the Yellowstone scene contained dips in height, the group's reflection changed as the mountains had become shadows. We also noticed that proctored exams the forest with the shrouded districts extends the timberland's classification misjudgement. In the most part, Random Forest conducts greatly with an immense power of preparatory tests. The Mississippi scene was granted with examples of homogeneity. This led to the high accuracy of the Random Forest, which surfaced every single existing classification [33]. We have different device computations for the class of sexual orientation but it is a basic endeavor to pick an incredible one. To choose the best calculation for the type of sexual inclination we conducted explorative see on device acing calculations. In this exploratory study of the gadget, we decided to break down in the process execution of various computations for sex class using voice data set. We concluded from this viewpoint that SVM and ANN are generating quality performance. ANN defeats SVM in the midst of tuning limits giving 99.87 per percent precision on test details. On speech dataset, vector backup machines and neural networks systems score better. Boundary tuning provides 98.6 percent SVM accuracy and 99.87 percent ANN accuracy. According to the above findings, we can conclude that deep neural systems function better in comparison with other gadgets calculating the economics of an individual 's gender orientation using auditory voice positions [34].

Machine Reading gives the restriction to the amazing and effective class of symbolism that has been remotely identified. The characteristics of a Machine trying to pick relevant stuff on incorporating the ability to handle elevated-dimensionality relevant data include space and delineating with highly enmeshed attributes. All things considered, approving a framework for gathering information. Description is not straightforward, and writing addresses several key problems with exhortations. This article thus provides an outline of the advantage from an applicable point of view of the device. We focus on guide vector machines, unwed choice trees (DTs), random forests, supported DTs, produced neural systems, and k-nearest neighbors (k-NN) around the provably fully developed systems. Issues that include calculation decision, preparation of data basic requirements, customer defined boundary selection and improvement, mark space factors that influence and reduce, and Machine learning expenses. We highlight such problems by trying to apply devices investigating class to different pairs

of insights that have been remotely identified publicly. On these lines, if potential, customers will search for the best approach with more than one classifier. A program incorporating the caret multipack bargain in R allows such a scene to be done quickly and efficiently, with no additional features in any scenario, if this inspection is well beyond the realm of the possible, there seems to be an agreement, by all accounts, that SVM, RF, and supported DTs are all-powerful procedures that seem to function properly under heaps of conditions. The search among these methodologies can come down to various variables, which are examined below. For starters, aided DTs are medium and are considerable along those lines less trying to engage when the set of data is massive and velocity is important. We endorse RF as it has been found to be generally strong to border settings. Using the default limit for factor scope and picking up a massive portion of trees (e. G.500) does seem to give a class accuracy close to what could be accomplished by simplifying. The method record sets used in this gander have been made publicly available through to the Machine learning Depository of Purdue University and University of California, Irvine (IUC). We may also choose to accept two anonymous pundits whose comments spurred the formulation [35].

A large part of the gadget training that considers centres around incremental slope mountaineering strategies and using nearby expertise to gather insights that are fundamental to area or global maxima. In this paper, along with a policy called against learning, we endorse the coaching of elective techniques for summing up to the performance achievable configuration. By using simple instructional strategies, understudies can boost in-depth data on the importance of authorization of dismissed truths from the training strategy, and that each issue requires its own techniques to be resolved. We are also the precondition for teaching adaptation using adequate realities by demonstrating that incredible bridge-approval granularities can be given remarkable results. Against acing is the situation where, regardless of how much you develop your estimates, the capital punishment of your contraction learned Approach is repeatable more unfortunate than the probability of speculating the appropriate response. So, for an inconvenience of two classes, if your variant is confidently under. The general concept wants to be offensive to learning, half accurate on secret information. Many other understudies, and even staff, are restricted to guide this scenario anyway it's clarification from the blue untruths in one of the least difficult, understanding used to show the need for a cloaked layer of discernment, tolerance investigation (e.g., How uptight are various datasets) [36].

Document function item selection is a fundamental and essential problem for text mining and retrieving information. Traditional. Traditional

Feature extraction methods involve handcrafted features. An efficient function is a lengthy feature to hand-design Deep learning, but aimed at new applications, allows new successful feature representation to be acquired. From data on instruction, Deep learning has made advances in text mining as a modern feature extraction technique. The key difference amongst deep learning and traditional approaches is that, instead of implementing handmade features, deep learning automatically discovers features from big data, which relies primarily on designers' previous knowledge and is extremely difficult to take advantage of big data. Deep learning, including millions of parameters, will automatically learn feature vectors from big data. This study outlines the typical methods used in extracting text features first and then expands freely. The creation of CNN (Convolution Neural Network) Recent years have received tremendous interest and a very high level of attention. Form of successful detection. Hubel and in the 1960s, Centered on studies into the visual cortex of the cat, Wiesel The notion of the receptive field[88] was put forward by cells. Fukushima, empowered, made neurocognitive predictions in the first CNN network deployment and also felt First of all, this crazy idea is introduced in the field of artificial [37]. The neural network, Then, in LeCun et alThe designing is dependent on the training of error gradient algorithms in the convolutional neural network, and the task set for some pattern recognition, the error gradient algorithm [38].

Traffic has been converted into the tough shape for structuring and adjusting uses through the vehicle's thought cycle of increasing reach. This situation has watched the issue of street mishaps, affected the accessible spa and the nation's budgetary system, and carried out the review on the issue’s response. Large tailored perspectives have expanded across the cycles of thought the creative carport upgrades and records with ease. The emergence of the need to increase information from these huge aligned measurements gained the data mining framework. In this investigation, a most ideal piece of technology was planned to become familiar with class techniques for calculating road missteps by using mining complexities. Scars in the street and miseries in the visible bruises of forming social orders are specific times and places. Controlling and coordinating visitors with advanced structures has emerged as a significant need in conjunction with the expansion of road malfunctions. Yes, only basic safety measures and through and unintended warnings spare you the mishaps in the road. It is mentioned

that hazard estimates and such damages should lessen by the orchestration and intercessions to be made as the after effects of the risks. Using contraction picking relevant stuff on is a systematic and thorough approach to track correct decisions with the knowledge to be made. AI assessments and estimates will carry logical methods and assumptions to the problem. Real injury information Contraction acing could be used routinely achieved. The required protection steps against capacity mishaps may be taken by revenge of the hazards to street wounds. The programming of the network, cell and measuring device that be computed by using certain equations for the mishap danger gauge. The guideline for even more prominent traffic feel can be provided with the guide to offer such applications to the personal and organizational use [39].

Chapter 2

Proposed Methodology

3.1 Introduction

This chapter explains the research methodology and its strategy of design. Purpose behind this research is to find out the impact of different classifiers on different data sets and after that compare them based on their performance on different parameters having on metrics. Initially, we have conducted an experiment with generic different datasets to find out their learning and then the result based on testing is carried out using our hybrid Approach having custom algorithms and riddled datasets for that. After that we conclude see the parameter labels (Features) and the behaviour of Approach with respect to data augmentation either fileting them or adding something to the original dataset, we concluded the result that changing in dataset will also tend to change in the results based on the techniques for particular hybrid Approach. We first collected data from the experiments and then the results of which are presented and analysed in the later chapters.

3.2 Research Method

We carried out an experiment on Multiple Datasets. Our experiment consists of several steps.

1. Data Collection (Dataset gathering)
2. Applying Deep Learning for data
3. Data Pre-Processing. (Removing unwanted data or Data cleaning for the features)
4. Identify the novelty of dataset
5. Preparing the dataset according to requirements.
6. Assigning classifiers for checking the performance and behaviour on dataset
7. Record the results
8. Visualize the Results for better understanding.
9. Compare the results of different Approach in metrics
10. Mark the best accurate results as leading.

11. Analysis on the experiments with existing knowledge-based results.

Figure 8 shows the Hybrid (Deep learning + Machine learning process) given below.

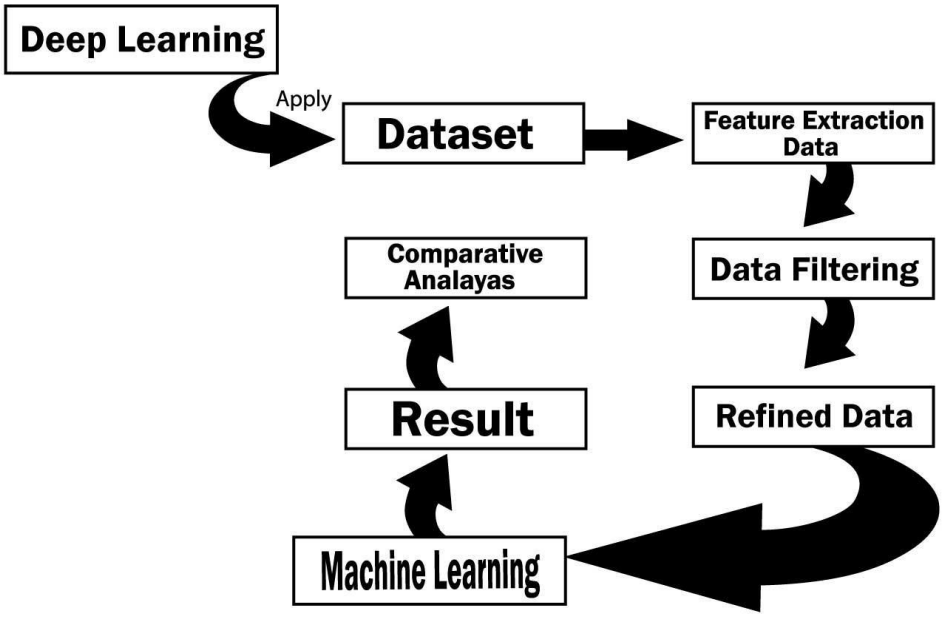


Figure 8 Hybrid (Deep learning Machine learning process)

Figure 9 shows the Machine learning process given below.

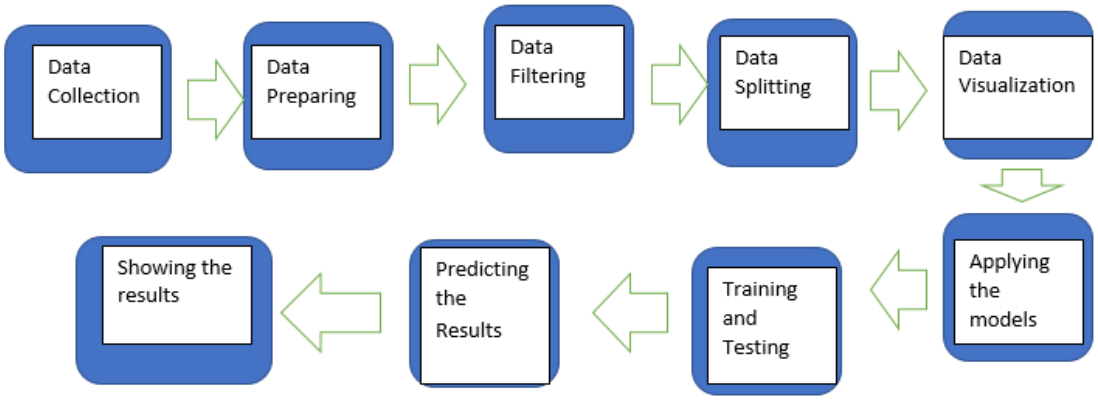


Figure 9 Machine learning process

Figure 10 shows the Enhanced Generic Approach Process given below.

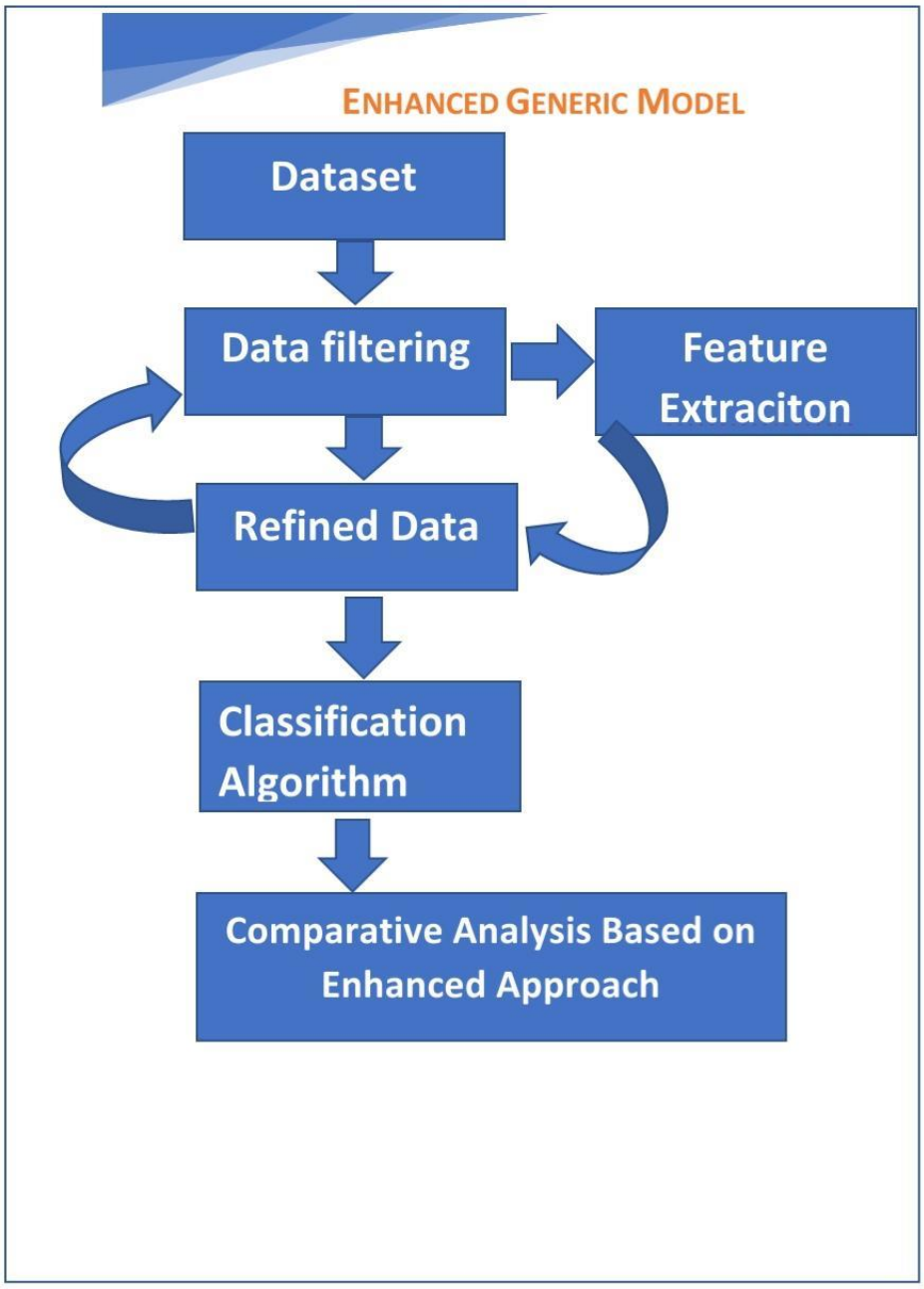


Figure 10 Enhanced Generic Approach Process (Approach 1)

Enhanced generic approach resembles the traditional approaches, but here we just did data pre-processing in a better way from raw dataset. Also feature selection is very important in terms of predictive analysis, so a significant dataset can be generated having a minimum impurity, for the better results and minimization of misclassification error. After that Machine learning techniques are applied and based on the parameters (precision, accuracy, recall and f1 score) recorded. After that a comparative analysis is

created to show the significant results in the performance of the classifiers among different datasets.

3.2.1 Detail overview of the experiment conducted

The experiment was conducted in two phases. In the first phase of experiment, we identified the different Datasets and Classifiers also ruined on generic Approach with some data pre-processing techniques for sample dataset to check the novelty. In the second phase, based on the filtered dataset received from the phase 1 from the original dataset, we created initial step by step experiments and then we carried out the all experiments and recorded them accordingly with the hybrid approach to achieve the maximum best results.

3.2.2 Experiment platform

In order to conduct our experiments we used tools PyCharm for local Machine and also cloud resources such as Jupyter co lab. These experiments are developed using Python, and for the dataset we used csv files. (We can use Jupyter co lab for implementation if your Machine Is slow or either your system is not able to bear work load for python anaconda etc.).

3.2.3 Factors

We have done the experiments using different classifiers based on the conclusion of result on different dataset, the factors that are encountered in experiments are:

1. Dataset Cleaning.
2. Identification of Data Labels.(Feature extraction)
3. Useful attributes selection.
4. Algorithm selection.
5. Sensitivity of Dataset.
6. Parameter for Results.
7. Training and Testing Ratio.
8. Results Output
9. Graphical Representation of Results.
10. Visualization of Results for accuracy

Figure 11 shows the Hybrid approach process given belows.

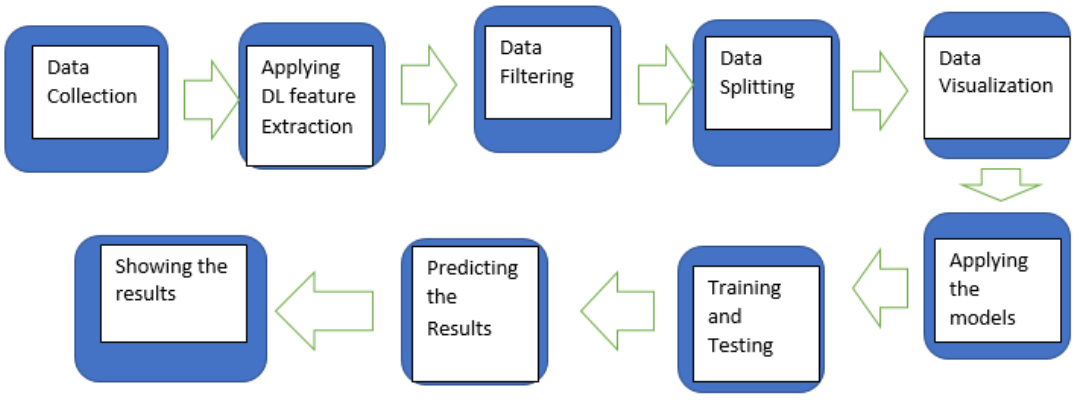


Figure 11 Hybrid approach process

Figure 12 shows the Proposed Hybrid Approach given below.

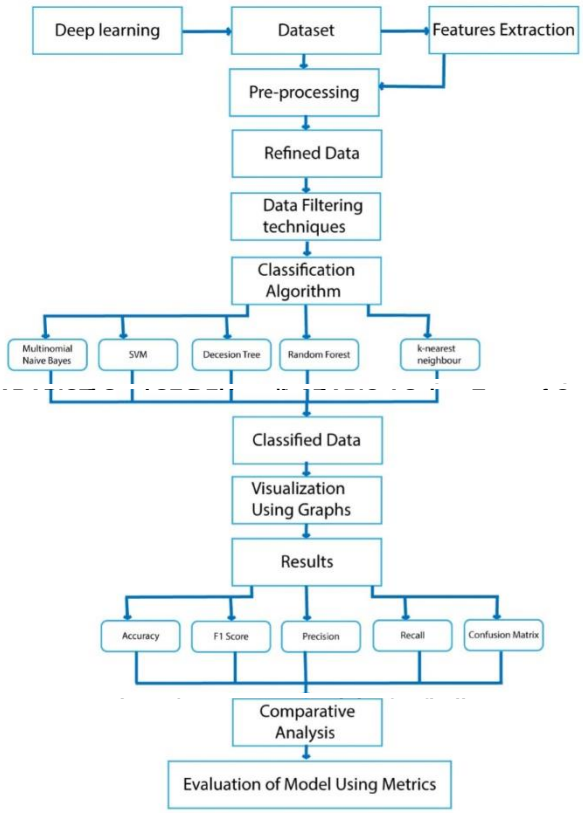


Figure 11 Proposed Hybrid Approach (Approach 2)

Figure 12 Proposed Hybrid Approach (Approach 2)

Deep Learning (DL) applied to the dataset with some starting layers of their approaches to get features after that these significant data is filtered and removes the unnecessary spaces or unwanted things, then this refined data is passed to the ML classifiers and after the classification ,the classified data is visualized using graphs can be seen at appendix ,after that based on the parameter for performance the precision ,accuracy, recall and f1 score recorded and a comparative analysis is made on different datasets.

3.3 Summary

In this chapter we had explained in detail the whole methodology of our research work and experiment that we had conducted experimental study by Machine learning classifiers using python programming. Experiment was held in two phases, first phase was about discovering dataset learning style and data pre-processing with features extractions and in the second phase we performed initial Classification and then recorded the results.

Chapter 3

Experiments and Results

4.1 Introduction

The size of knowledge generated by various associations is increasing step by step exponentially. In web-based social networking, open and private evaluations of different topics or issues are communicated. Twitter is a small-scale blogging platform that enables individuals to express and articulate their points of view, or send comments. Inquiry report on Facebook is a means of dissecting the customer's feeling by tweeting details (tweets). Analysis investigation refers to the use of natural language management, text analysis, and controlling electronic sensation investigation. Assumption Analysis also known as Opinion Mining (OM) aims at discovering the thoughts, mentalities and attitudes of individuals about a Stuff. The factor will talk to individuals, and multiple occasions. It gives us data on the positive, negative or unbiased extreme point and is the subject of an assumption.

4.2 Experiment on Amazon Product Reviews Dataset

Steps for Data Labelling

In this phase we clean the data as well as label the data

- 0 for Poor
- 1 for Neutral
- 2 for Good

Read the data from the dataset file 'Amazon_Unlocked_Mobile.csv' and then add new column for the labels to use with.

Data Cleaning

Remove all the empty/ vacant rows containing blank cells (Spaces).The subsequent data is stored as 'labelled_dataset.csv

Data Pre-Processing Technique

The following data pre preprocessing techniques are applied to alter raw reviews (data) to cleaned review (data), thus that it will become easier for us to ensure feature extraction in the next step.

- By removing html tags using Beautiful Soup
- by removing the non-character like digits or symbols
- By converting them to lower case
- By removing stop words such as "the" or "and" if needed
- By converting to roots of the words through stemming if needed

Bag Of Words (Counter)

The sentiment analysis of given text can be done in two ways. First, we need to find a word entrenching to convert a text into a numerical representation. Second, we fit the numerical representations of text to Machine learning algorithms or deep learning architectures. In our case we only use Machine learning, but deep learning can also be used in future work.

One common method to word transplanting is frequency-based enclosure such as Approach Bag of Words (Bow). Bow Approach can learn a comprehension list from a given amount and reflects each file on the basis of certain word counting methodologies. In this part, we will explore the Approach performance of using Bow with supervised learning algorithms. Here's the workflow steps are mentioned in this part.

Step 1: Data Pre-processing cleaned reviews (Imbalanced dataset) or raw reviews

Step 2: Create Bow (word bag) using the vectorizer Count / TF Idf in Sklearn.

Step 3: Convert text of the analysis into numerical representations (function vectors)

Step 4: Adapt vectors to supervised learning algorithms (e.g. Naive Bayes, Logistic Regression, etc.)

4.2.1 Multinomial Naive Bayes on Amazon Product Reviews

In Table 3 shows the Results against Multinomial Naive Bayes on Amazon Product Reviews

Table 3 Results against Multinomial Naive Bayes on Amazon Product Reviews

Sr:No	Approachs Used	Precision	Recall	Accuracy	F1score
1.	Multinomial Naive Bayes	0.84	0.86	0.86	0.85

4.2.2 Logistic Regression on Amazon Product Reviews

In Table 4 shows Results against Logistic Regression on on Amazon Product Reviews

Table 4 Results against Logistic Regression on on Amazon Product Reviews

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	Logistic Regression	0.87	0.88	0.88	0.86

4.2.3 SVM on Amazon Product Reviews on

In Table 5 shows the Results against SVM on Amazon Product Reviews

Table 5 Results against SVM on Amazon Product Reviews

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1score
1	SVM	0.94	0.93	0.94	0.93

4.2.4 Random forest on Amazon Product Reviews

In Table 6 shows the Results against Random forest on Amazon Product Reviews.

Table 6 Results against Random forest on Amazon Product Reviews

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1score
1.	Random forest	0.93	0.86	0.93	0.93

4.2.5 Decision Tree on Amazon Product Reviews

In Table 7 shows the Results against Decision Tree on Amazon Product Reviews

Table 7 Results against Decision Tree on Amazon Product Reviews

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1score
1.	Decision tree	0.92	0.88	0.92	0.93

4.2.6 K-Nearest Neighbour on Amazon Product Reviews

In Table 8 shows the Results against K-Nearest Neighbour 0n on Amazon Product Reviews

Table 8 Results against K-Nearest Neighbour 0n on Amazon Product Reviews

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	K-nearest neighbour	0.96	0.93	0.96	0.95

4.2.7 Summary of approaches used on Amazon Product Reviews dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	Multinomial Naive Bayes	0.84	0.86	0.86	0.85
2.	Logistic Regression	0.87	0.88	0.88	0.86
3.	SVM	0.94	0.93	0.94	0.93
4.	Random forest	0.93	0.86	0.93	0.93
5.	Decision tree	0.92	0.88	0.92	0.93
6.	K-nearest neighbour	0.96	0.93	0.96	0.95

4.3 Experiment on Breast Cancer Dataset

Goal of the ML project

Breast cancer is a risky condition for women. If it fails to detect in the preliminary phase then the patient 's death will occur. It is a common cause of death in women all over the world. Nearly 12 percent of women suffer from cancer internationally and the figure continues to grow.

Doctors don't recognise every single patient with breast cancer. That's why I am an Apprenticeship Engineer . Data Scientists enter the picture as they have awareness of maths and computing capacity. We have extracted characteristics of cells transfected by breast tumors and normal cells of people. As a Machine learning engineer / data scientist, to distinguish malignant and benign tumour, an ML Approach must be developed. To complete this ML project we have used the classification technique for supervised Machine learning.

To build the best Approach, we have to train and test the dataset with multiple Machine learning algorithms then we find the best ML Approach.

4.3.1 Multinomial Naive Bayes on Breast Cancer dataset

In Table 9 shows the result Against Multinomial Naive Bayes on Breast Cancer dataset Reviews

Table 9 result Against Multinomial Naive Bayes on Breast Cancer dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	Multinomial Naive Bayes	0.97	0.96	0.97	0.96

4.3.2 Logistic Regression on Breast Cancer dataset

In Table 10 shows the Result against Logistic Regression on Breast Cancer dataset Reviews

Table 10 Result against Logistic Regression on Breast Cancer dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1score
1.	Logistic Regression	0.93	0.92	0.94	0.94

4.3.3 SVM on Breast Cancer dataset

In Table 11 shows the results against SVM on Breast Cancer dataset

Table 11 results against SVM on Breast Cancer dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1	SVM	0.91	0.92	0.93	0.92

4.3.4 Random forest on Breast Cancer dataset

In Table 12 shows the results against SVM on Breast Cancer dataset

Table 12 results against SVM on Breast Cancer dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	Random forest	0.98	0.97	0.96	0.97

4.3.5 Decision Tree on Breast Cancer dataset

Table 13 shows the results against Decision Tree on Breast Cancer dataset.

Table 13 results against Decision Tree on Breast Cancer dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	Decision tree	0.96	0.94	0.97	0.95

4.3.6 K-Nearest Neighbour on Breast Cancer dataset

In Table 14 shows the Result against K-Nearest Neighbour on Breast Cancer dataset.

Table 14 Result against K-Nearest Neighbour on Breast Cancer dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	K-nearest neighbour	0.94	0.92	0.93	0.93

4.3.7 Summary of approaches used on Breast Cancer dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	Multinomial Naive Bayes	0.97	0.96	0.97	0.96
2.	Logistic Regression	0.93	0.92	0.94	0.94
3.	SVM	0.91	0.92	0.93	0.92
4.	Random forest	0.98	0.97	0.96	0.97
5.	Decision tree	0.96	0.94	0.97	0.95
6.	K-nearest neighbour	0.94	0.92	0.93	0.93

4.4 Experiment on Amazon Products Reviews Dataset

This Dataset comprises 40,000 Amazon customer reviews, star ratings, date of review, variant and feedback of various amazon products like Alexa Echo, Echo dots.The objective is to discover insights into consumer reviews and perform sentiment analysis on the data.

4.4.1 Multinomial Naive Bayes on Amazon Products Reviews dataset

In Table 15 shows the Result against Multinomial Naive Bayes on Amazon Products Reviews.

Table 15 Result against Multinomial Naive Bayes on Amazon Products Reviews dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1score
1.	Multinomial Naive Bayes	0.93	0.92	0.92	0.88

4.4.2 Logistic Regression on Amazon Products Reviews dataset

In Table 16 shows the Result against Logistic Regression on Amazon Products Reviews dataset.

Table 16 Result against Logistic Regression on Amazon Products Reviews dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	Logistic Regression	0.92	0.92	0.92	0.88

4.4.3 SVM on Amazon Products Reviews dataset

Table 17 shows the Result against Logistic Regression on Amazon Products Reviews dataset.

Table 17 Result against Logistic Regression on Amazon Products Reviews dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1	SVM	0.94	0.94	0.94	0.92

4.4.4 Random forest on Amazon Products Reviews dataset

Table 18 shows the result against Random forest on Amazon Products Reviews dataset.

Table 18 result against Random forest on Amazon Products Reviews dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	Random forest	0.96	0.95	0.96	0.94

4.4.5 Decision Tree on Amazon Products Reviews dataset

In Table 19 shows the Result against Decision Tree on Amazon Products Reviews dataset.

Table 19 Result against Decision Tree on Amazon Products Reviews dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	Decision tree	0.93	0.92	0.93	0.92

4.4.6 K-Nearest Neighbour on Amazon Products Reviews dataset

In Table 20 shows the Result against K-Nearest Neighbour on Amazon Products Reviews.

Table 20 Result against K-Nearest Neighbour on Amazon Products Reviews dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	K-nearest neighbour	0.91	0.92	0.91	0.91

4.4.7 Summary of approaches used on Amazon Products Reviews dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score

1.	**Multinomial Naive Bayes**	**0.93**	**0.92**	**0.92**	**0.88**
2.	**Logistic Regression**	**0.92**	**0.92**	**0.92**	**0.88**
3.	**SVM**	**0.94**	**0.94**	**0.94**	**0.92**
4.	**Random forest**	**0.96**	**0.95**	**0.96**	**0.94**
5.	**Decision tree**	**0.93**	**0.92**	**0.93**	**0.92**
6.	**K-nearest neighbour**	**0.91**	**0.92**	**0.91**	**0.91**

4.5 Experiment on the Bilingual Hindi-English Social Media Dataset

Bilingual Sentiment Analysis is a subset of multilingual sentiment analysis which is one of many popular types of sentiment analysis. It handles the sentiment analysis of text that includes more than one language.

Dataset

The dataset that we used, obtained from "Kaggle". It contains 1,600,000 tweets.

Steps for Data Labelling

In this phase we clean the data as well as label the data

- 0 for negative
- 2 for neutral
- 4 for positive

Data Cleaning

Remove all the empty/ vacant rows containing blank cells. . Remove noise contains in a dataset in the form of URLs, stop words, special characters, or white spaces in tweets.

4.5.1 Multinomial Naive Bayes

Table 21 shows the Result against Multinomial Naive Bayes on Bilingual dataset.

Table 21 Result against Multinomial Naive Bayes on Bilingual dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	**Multinomial Naive Bayes**	**0.64**	**0.62**	**0.62**	**0.61**

4.5.2 Logistic Regression

Table 22 shows the Result against Logistic Regression on Bilingual dataset.

Table 22 Result against Logistic Regression on Bilingual dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	Logistic Regression	0.64	0.62	0.62	0.61

4.5.3 SVM

Table 23 shows the Result against SVM on a Bilingual dataset.

Table 23 Result against SVM on Bilingual dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1	SVM	0.64	0.62	0.62	0.61

4.5.4 Random forest

Table 24 shows the Result against Random forest on Bilingual dataset.

Table 24 Result against Random forest on Bilingual dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	Random forest	0.64	0.62	0.62	0.61

4.5.5 Decision Tree

In Table 25 shows the Result against Decision tree on Bilingual dataset.

Table 25 Result against Decision tree on Bilingual dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1score
1.	Decision tree	0.64	0.61	0.61	0.59

4.5.6 K-Nearest Neighbour

In Table 26 shows the Result against K-Nearest Neighbor on Bilingual dataset.

Table 26 Result against K-Nearest Neighbor on Bilingual dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	K-nearest neighbour	0.63	0.62	0.61	0.60

4.5.7 Summary of approaches used on Bilingual dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	Multinomial Naive Bayes	0.64	0.62	0.62	0.61
2.	Logistic Regression	0.64	0.62	0.62	0.61
3.	SVM	0.64	0.62	0.62	0.61
4.	Random forest	0.64	0.62	0.62	0.61
5.	Decision tree	0.64	0.61	0.61	0.59
6.	K-nearest neighbour	0.63	0.62	0.61	0.60

4.6 Experiment on Drug Analysis using sentiment Technique

Machine learning can be used as COVID-19 prediction in another way.

Steps for Data Labelling

In this phase we clean the **data** as well as **label** the data

- 0 for Negative
- 1 for Positive

Data Cleaning

Remove all Noise contained in a dataset in the form of URLs, stop words, special characters, or white spaces.

4.6.1 Multinomial Naive Bayes

Table 27 shows the Result Against Multinomial Naive Bayes on Drug dataset.

Table 27 Result Against Multinomial Naive Bayes on Drug dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	Multinomial Naive Bayes	0.92	0.94	0.92	0.92

4.6.2 Logistic Regression

Table 28 shows the Result Against Logistic Regression on Drug dataset.

Table 28 Result Against Logistic Regression on Drug dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score

1.	**Logistic Regression**	**0.93**	**0.93**	**0.94**	**0.94**

4.6.3 SVM

Table 29 shows the Result Against SVM on Drug dataset.

Table 29 Result Against SVM on Drug dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1	**SVM**	**0.91**	**0.92**	**0.92**	**0.91**

4.6.4 Random forest

Table 30 shows the Result Against Random forest on Drug dataset.

Table 30 Result Against Random forest on Drug dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	**Random forest**	**0.94**	**0.94**	**0.93**	**0.94**

4.6.5 Decision Tree

In Table 31 shows the Result Against Decision tree on Drug dataset.

Table 31 Result Against Decision tree on Drug dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	**Decision tree**	**0.91**	**0.92**	**0.91**	**0.92**

4.6.6 K-Nearest Neighbour

In Table 32 shows the Result Against K-Nearest Neighbor on Drug dataset.

Table 32 Result Against K-Nearest Neighbor on Drug dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score
1.	**K-nearest neighbour**	**0.92**	**0.93**	**0.93**	**0.93**

4.6.7 Summary of approaches used on Drug Analysis dataset

Sr:No	Approaches Used	Precision	Recall	Accuracy	F1 Score

1.	**Multinomial Naive Bayes**	**0.92**	**0.94**	**0.92**	**0.92**
2.	**Logistic Regression**	**0.93**	**0.93**	**0.94**	**0.94**
3.	**SVM**	**0.91**	**0.92**	**0.92**	**0.91**
4.	**Random forest**	**0.94**	**0.94**	**0.93**	**0.94**
5.	**Decision tree**	**0.91**	**0.92**	**0.91**	**0.92**
6.	**K-nearest neighbour**	**0.92**	**0.93**	**0.93**	**0.93**

4.7 Comparative Analysis

We have presented dataset wise results for all the techniques in the above section. Here we are presenting a comparative analysis on all the datasets with an enhanced generic approach. Comparative Analysis on different dataset with Enhanced Generic Approach is shown in table 33 given below.

Table 33 Comparative Analysis on different dataset with Enhanced Generic Approach

Sr No	About dataset	Approachs Used	Precision	Recall	Accuracy	F1score
1.	Amazon mobile Reviews	1. Multinomial Naive Bayes	0.84	0.86	0.86	0.85
		2. Logistic Regression	0.87	0.88	0.88	0.86
		3. SVM	0.94	0.93	0.94	0.93
		4. Random forest	0.93	0.94	0.93	0.93
		5. Decision tree	0.92	0.93	0.92	0.93
		6. K-nearest neighbour	**0.96**	**0.95**	**0.96**	**0.95**
2.	Breast Cancer Detection Data	1. Random forest	0.97	0.96	**0.97**	0.96
		2. Decision tree	0.93	0.92	0.94	0.94
		3. Naïve Bayes	0.91	0.92	0.93	0.92
		4. SVM	**0.98**	**0.97**	0.96	**0.97**
		5. Logistic Regression	0.96	0.94	0.97	0.95

		6. K-nearest neighbour	0.94	0.92	0.93	0.93
3.	Amazon Alexa Products	1. Naïve Bayes	0.93	0.92	0.92	0.88
		2. Logistic Regression	0.92	0.92	0.92	0.88
		3. Random forest	0.94	0.94	0.94	0.92
		4. SVM	**0.96**	**0.95**	**0.96**	**0.94**
		5. Decision Tree	0.93	0.92	0.93	0.92
		6. K-nearest neighbour	0.91	0.92	0.91	0.91
4.	Bilingual "Hindi-English" Data	1. Naïve Bayes	0.64	0.62	0.62	0.61
		2. K-nearest neighbour	0.64	0.62	0.62	0.61
		3. Random forest	0.64	0.62	0.62	0.61
		4. Logistic Regression	**0.64**	**0.62**	**0.62**	**0.61**
		5. Decision Tree	0.64	0.61	0.61	0.59
		6. SVM	0.63	0.62	0.61	0.60
5	Drugs Record Dataset	1. Random forest	0.92	0.91	0.92	0.92
		2. Decision tree	0.93	0.93	**0.94**	0.94
		3. Naïve Bayes	0.91	0.92	0.92	0.91
		4. SVM	**0.94**	**0.93**	0.93	**0.94**
		5. Logistic Regression	0.91	0.92	0.91	0.92
		6. K-nearest neighbour	0.92	0.92	0.93	0.93

Figure 13 shows the Comparative Metrics for Performance of Amazon mobile reviews dataset given below.

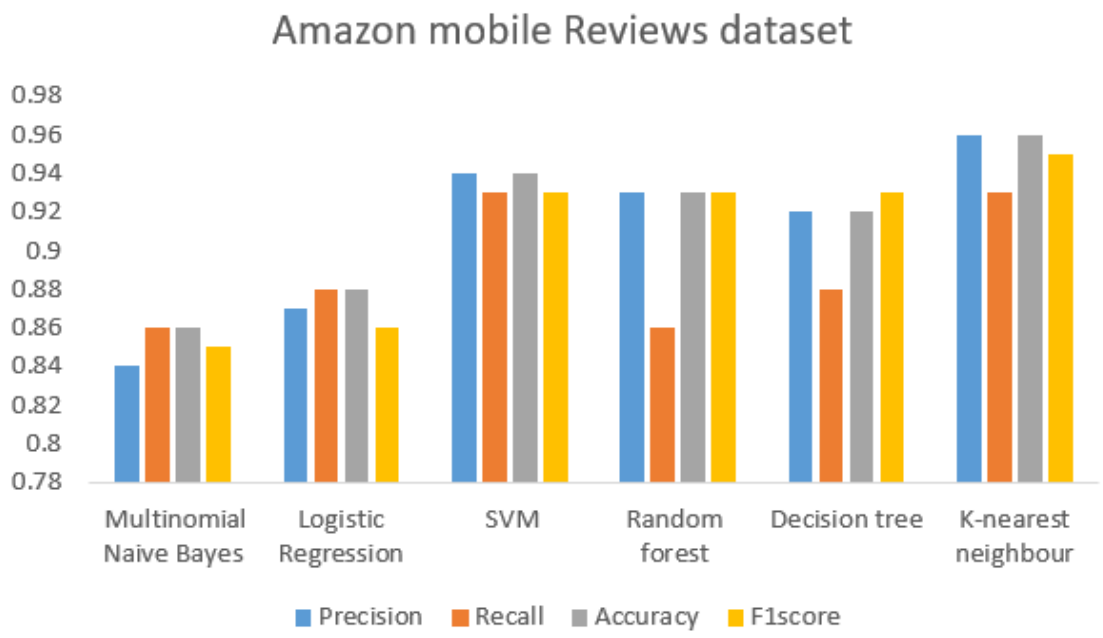


Figure 13 Comparative Metrics for Performance of Amazon mobile reviews dataset

Figure 14 shows the Comparative Metrics for Performance of breast cancer dataset given below.

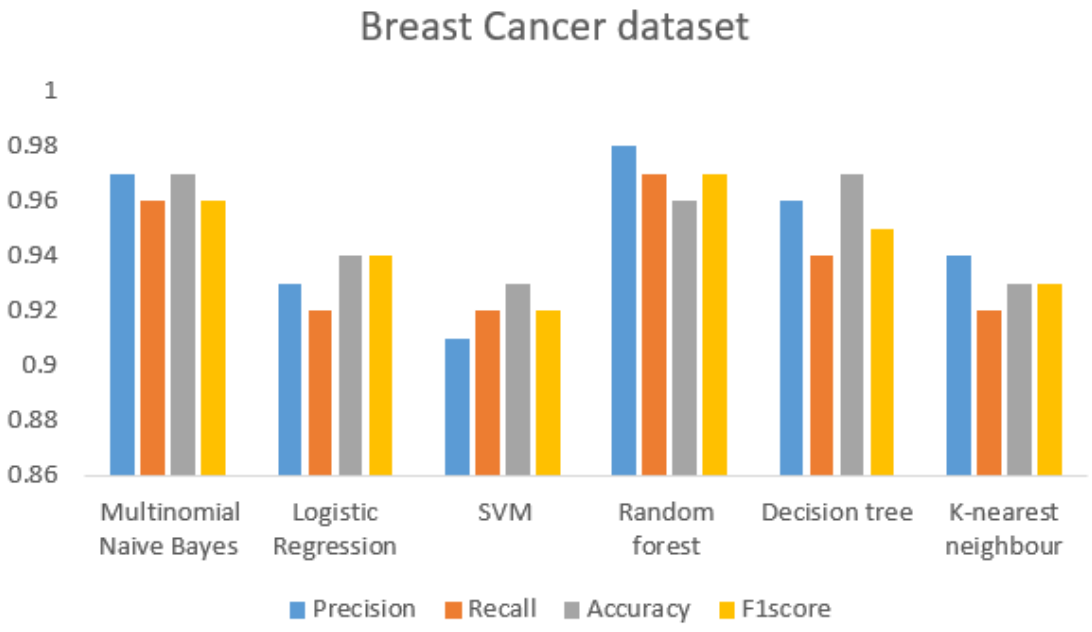


Figure 14 Comparative Metrics for Performance of breast cancer dataset

Figure 15 shows the Comparative Metrics for Performance of Amazon Products reviews dataset given below.

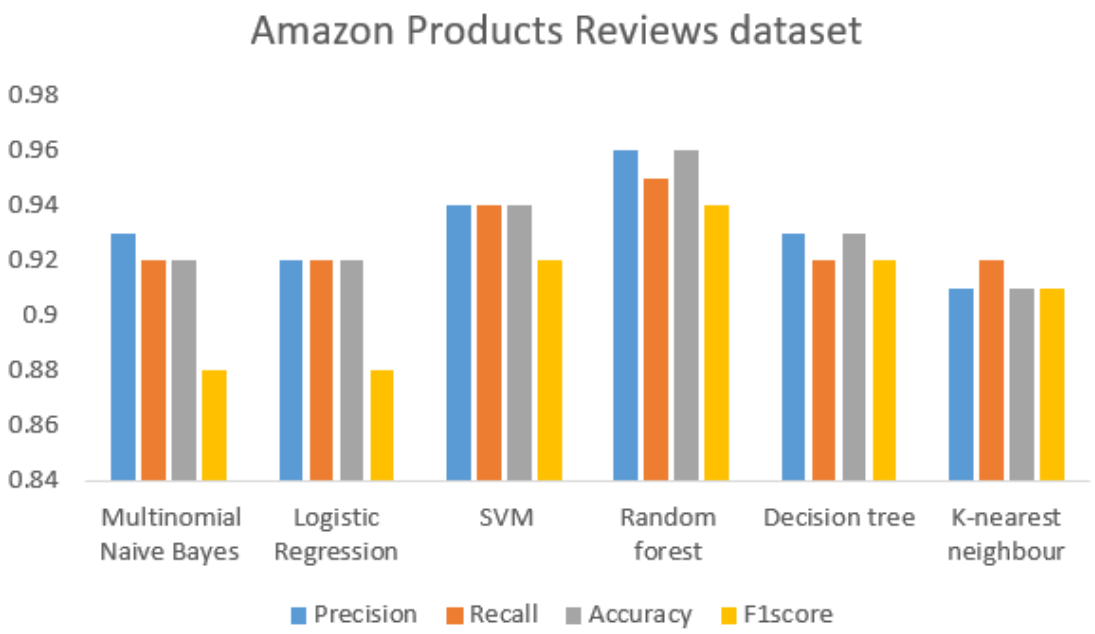


Figure 15 Comparative Metrics for Performance of Amazon Products reviews dataset

Figure 16 shows the Comparative Metrics for Performance of Bilingual dataset given below.

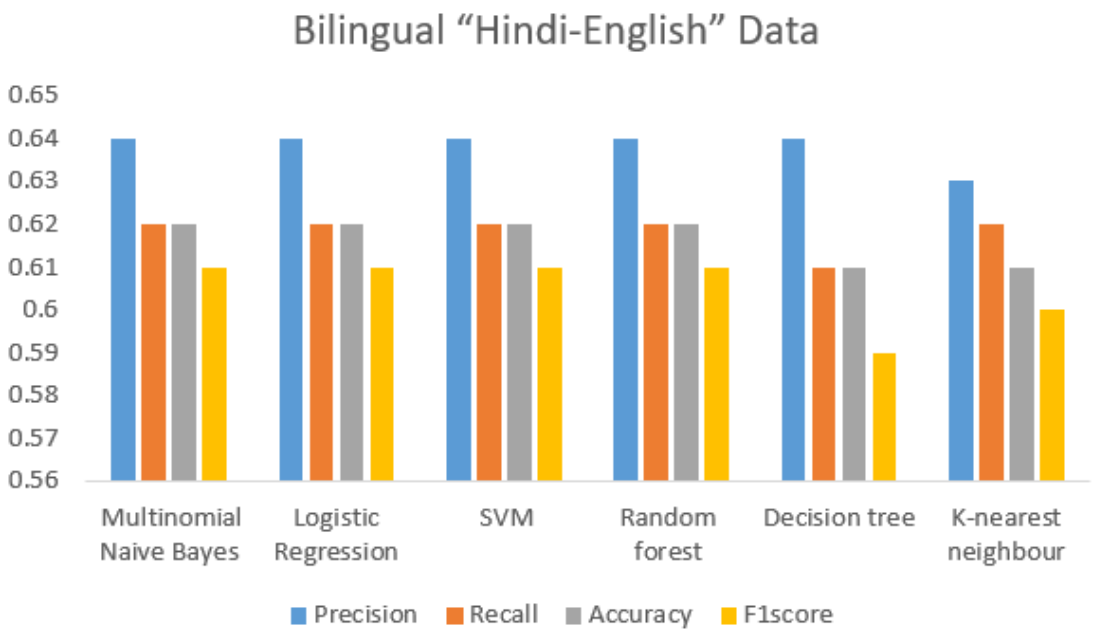


Figure 16 Comparative Metrics for Performance of Bilingual dataset

Figure 17 shows the Comparative Metrics for Performance of Drugs record dataset given below.

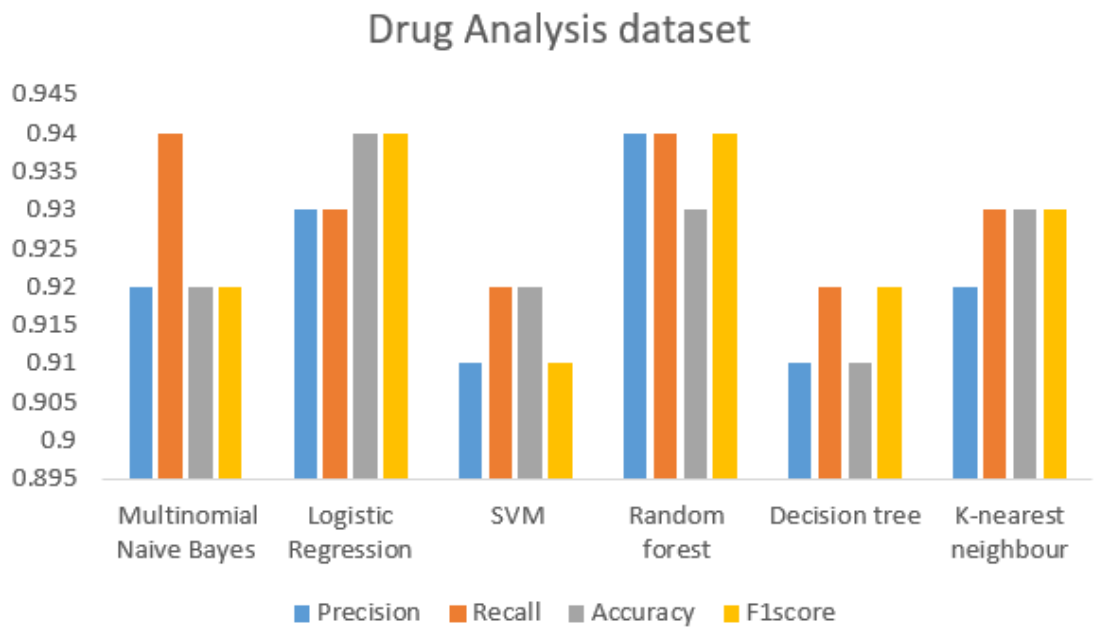


Figure 17 Comparative Metrics for Performance of Drugs record dataset

Now with the help of the hybrid approach proposed, we are presenting a comparative analysis on all the datasets. Here we can see that our technique is enhancing the performance parameters. Comparative Analysis on different dataset with Hybrid approach is shown in table 34 given below.

Table 34 Comparative Analysis on different dataset with Hybrid approach

Sr No	About dataset	Approachs Used	Precision	Recall	Accuracy	F1score
1.	Amazon mobile Reviews	1. Multinomial Naive Bayes	0.85	0.87	0.87	0.88
		2. Logistic Regression	0.88	0.88	0.88	0.87
		3. SVM	0.95	0.95	0.94	0.95
		4. Random forest	0.94	0.94	0.95	0.95
		5. Decision tree	0.94	0.93	0.92	0.94
		6. K-nearest neighbour	**0.97**	**0.96**	**0.96**	**0.97**
2.	Breast Cancer Dataset	1. Multinomial Naive Bayes	**0.98**	**0.97**	**0.98**	0.97
		2. Logistic Regression	0.94	0.93	0.94	0.93
		3. SVM	0.92	0.92	0.93	0.92
		4. Random forest	**0.98**	**0.97**	0.97	**0.98**
		5. Decision tree	0.96	0.95	**0.98**	0.96
		6. K-nearest neighbour	0.94	0.92	0.95	0.94
3.	Amazon Alexa Products	1. Multinomial Naive Bayes	0.93	0.94	0.94	0.88
		2. Logistic Regression	0.93	0.93	0.94	0.89
		3. SVM	0.95	**0.95**	0.96	0.93
		4. Random forest	**0.96**	**0.95**	**0.97**	**0.95**
		5. Decision tree	0.93	0.94	0.94	**0.95**
		6. K-nearest neighbour	0.91	0.92	0.93	0.94
4.	Bilingual “Hindi-English” Data	1. Multinomial Naive Bayes	0.69	0.68	0.69	0.68
		2. Logistic Regression	0.64	0.63	0.64	0.62
		3. SVM	**0.70**	**0.70**	**0.70**	**0.70**
		4. Random forest	0.67	0.67	0.66	0.67

		5. Decision tree	0.69	0.68	0.69	0.68
		6. K-nearest neighbour	0.66	0.65	0.66	0.65
5	Drugs record Data	1. Multinomial Naive Bayes	0.94	0.94	0.93	0.93
		2. Logistic Regression	0.94	0.94	**0.95**	**0.96**
		3. SVM	0.92	0.92	0.92	0.91
		4. Random forest	**0.95**	**0.94**	0.94	0.94
		5. Decision tree	0.91	0.92	0.91	0.92
		6. K-nearest neighbour	0.94	0.93	0.93	0.94

So, we can conclude that with the help of our proposed mechanism, the performance is increased and parameters show good results in terms of Precision, Accuracy, Recall and F1 score as compared to the traditional and enhanced generic approach. Therefore, we can say that the hybrid approach will give best performance in future as shown in above table. Comparative analysis shows clear enhanced results based on the parameters for different dataset which classifier will increase the performance among all classifiers. Figure 18 shows the Comparative Metrics for Performance of Amazon mobile reviews dataset given belows.

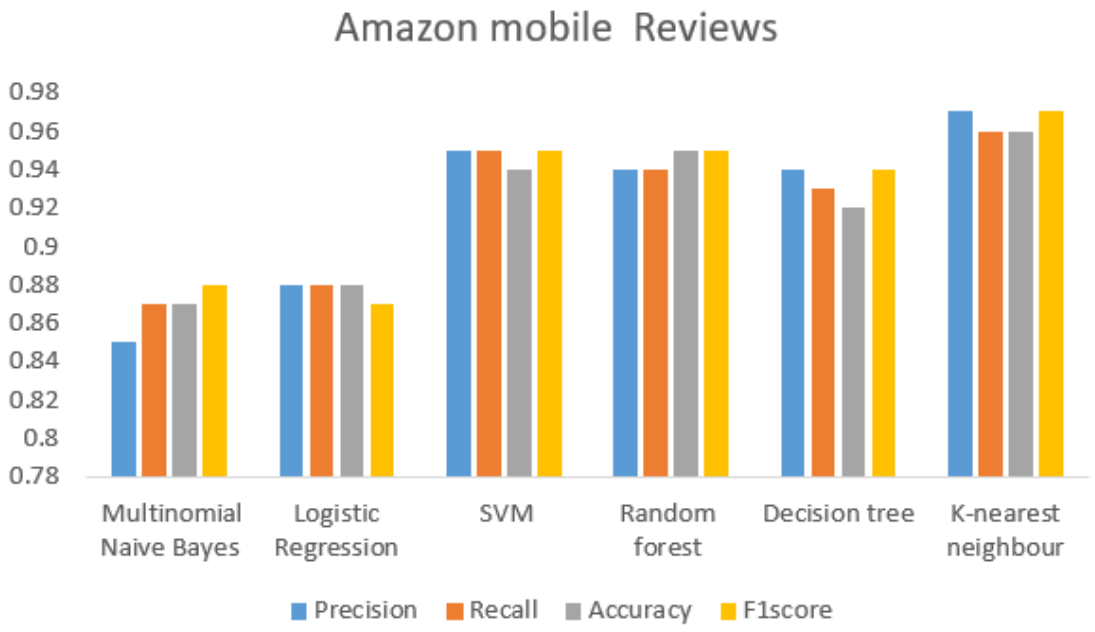


Figure 18 Comparative Metrics for Performance of Amazon mobile reviews dataset

Figure 19 shows the Comparative Metrics for Performance of breast cancer dataset given belows.

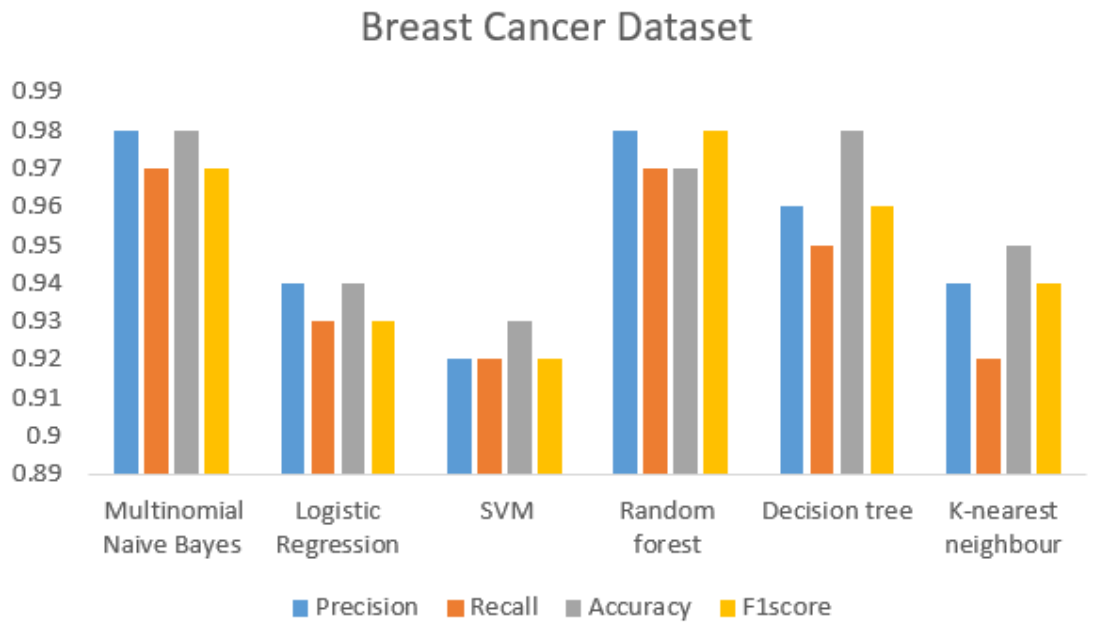


Figure 19 Comparative Metrics for Performance of breast cancer dataset

Figure 20 shows the Comparative Metrics for Performance of Amazon Products reviews dataset given belows.

Figure 20 Comparative Metrics for Performance of Amazon Products reviews

Figure 21 shows the Comparative Metrics for Performance of Bilingual dataset given belows.

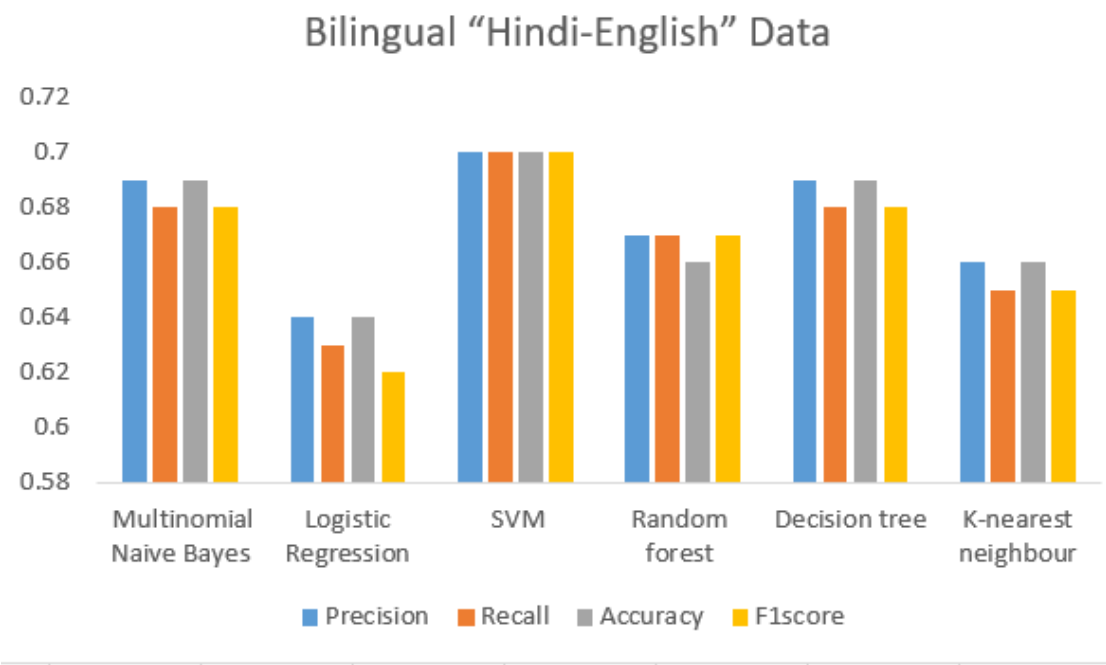


Figure 21 Comparative Metrics for Performance of Bilingual dataset

Figure 22 shows the Comparative Metrics for Performance of Drugs record dataset given below.

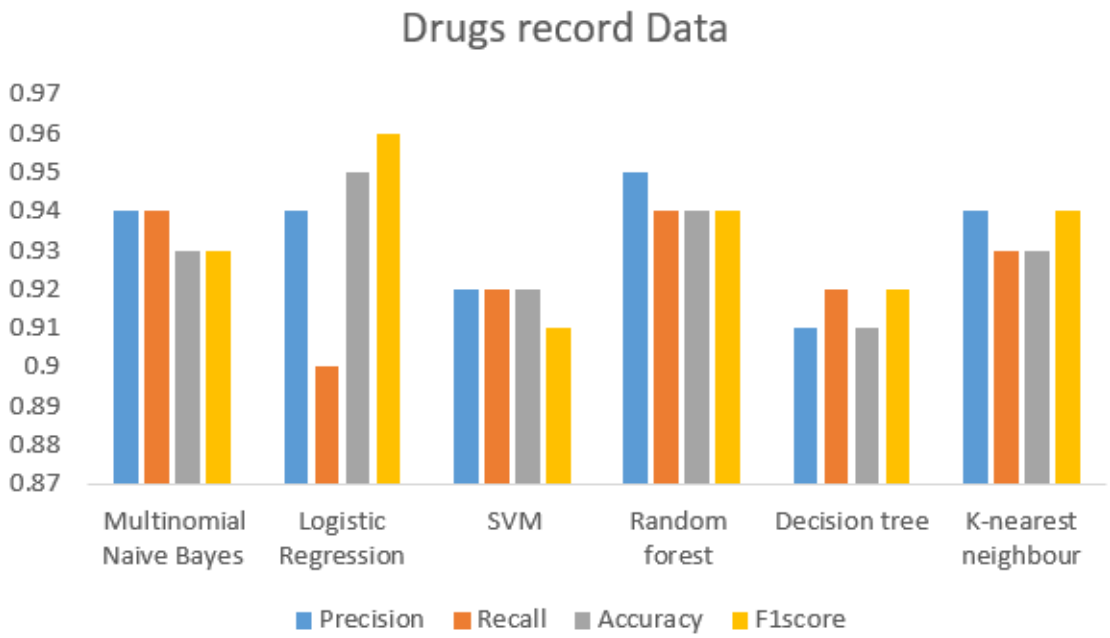


Figure 22 Comparative Metrics for Performance of Drugs record dataset

Chapter 4

Conclusion and Future Work

5.1 Conclusion

There is a lot of work in Machine learning nowadays because it's emerging day by day and very demanding in terms to solve predictive analysis. Scenarios where prediction is required among different areas. Analysis is dependent on the group of individuals for the selection of algorithms or classifiers used for the novelty of a dataset that is responsible for better results in less time. This is the most challenging task to do. Researchers have done a lot of work in this area, but still, there is a scope for Machine learning techniques for the comparative aspects in the formation of hybrid approach for the different classifiers showing the different results based on the datasets. Aim of this research is to provide a new technique for feature extraction and then selection based on data pre-processing techniques based on the deep learning classifier for best filtered data extractions for different selection of classifiers having the different results based on the pre-processing techniques and after applying specific algorithm as per the literature and knowledge to see the novelty-based results. We have focused on identifying the features and attributes of different dataset and then compare them based on their results that affect dataset formation to more accuracy.

Understandability is subjectively determined by the form of learning process and the scale of the resulting representation of the information. Hybrid methods increase the aspects ratio for improvements in predictive accuracy over standard understandable methods using generic Approach. Our hybrid Approach does classification using feature extraction in pre-processing of raw dataset by normalization for better classification, by removing unwanted noise from dataset either in the form of features or Data, to improve accuracy and minimization of Error Rate for the smaller amount of training of significant dataset and lesser amount of training time. The hybrid comparative analysis needed for the selection of dataset with respect to Machine learning classifiers that will have different results with different data sets. Input with maximum accurate results are reproduced from our hybrid approach. After that A comparative Analysis of different algorithms with different/or same datasets can be made and that tells us about the accuracy and rate of misclassification Error. We Infer

the results based on the different experiments conducted ,if the raw data is being filtered and selective usable features are extracted so the accuracy will be increased based on the circumstances depending upon the novelty of the dataset. A comparative Analysis shows the different results among the datasets for precision, accuracy, recall and f1 score against the generic techniques that are being modified by data pre-processing techniques for optimal results and also for the hybrid Approach for the best output result among them all.

With the help of this algorithm mentioned in the paper, we can effectively train our ML and DL models with more accuracy and less training time. This algorithm filters the dataset and divides the dataset into different classes based on their features and behaviours. This classification of dataset plays a key role in improving our model. It uses different techniques of both machine learning and deep learning to classify and analyse the dataset. It works in the form of a pipeline. It takes in the data, analyses the data, applies some feature engineering on the data, and returns data in its most refined form. Then it applies different machine learning and deep learning techniques on that dataset, and after that returns the model with highest accuracy and minimum training time.

5.2 Future work

Our research work can be extended in the following ways

1. Include open ended questions in our experiment and use Deep Learning Approaches (to find out the impact of results on large datasets).
2. Use Multiple Classifiers against different parameters attributes e.g., either for sentiment analysis to find out the results from previous history of any different or specific dataset.
3. Integrate this research study with deep learning approaches. So, that DL can handle multiple and large dataset or either compare them and made comparative analysis study by doing experiments on different dataset.
4. dataset.
5. Use Deep Learning for feature selection and extraction e.g., from images dataset e.g. Covid-19 X-rays for lungs and then apply Machine learning algorithm/Classifiers also can be compared with deep learning Approach.

5.3 Limitations

The Limitations of this research are as under:

1. For the significant dataset , we used English dataset and roman dataset only.
2. Image based Dataset may show different results depending upon the 2d layer deep learning and Approach, or may be large dataset may need a GPU based systems for the better and optimal output in a small amount of time.
3. Different datasets may show different results, check the datapreproessing techniques and datatypes with respect to the requirements.

References

[1] C. Sammut & g. I. Webb, "encyclopedia of Machine learning and data mining", springer, 2017

[2] Dalibor bužić *, jasminka dobša **, lyrics classification using naive bayes , mipro 2018.

[3] Ayman m mansou , texture classification using naïve bayes classifier, ijcsns international journal of computer science and network security, 2018

[4] Ianping gou and wenmo qiu, a local mean representation-based k-nearest neighbor classifier jjiangsu university, jiangsu key laboratory of security technology for industrial cyberspace, china. 2019

[5] Biau gerard and gerard biau. Journal of Machine learning research 13 analysis of a random forests Approach , universite pierre et marie curie – paris 2012

[6] Amit gupte, sourabh joshi, pratik gadgul, akshay kadam, amit gupte et al , comparative study of classification algorithms used in sentiment analysis, / (ijcsit) international journal of computer science and information technologies, 2014

[7] Saeed banihashemia*, grace dinga, jack wangb, developing a hybrid Approach of prediction and classification algorithms for building energy consumption , 1st international conference on energy and power, icep2016, rmit university, melbourne, australia 2017

[8] Nafizatus salmi and zuherman rustam , naïve bayes classifier Approachs for predicting the colon cancer , iop conference series: materials science and engineering, 2019

[9] B. Lantz, "Machine learning with r", packt publishing ltd, 2015.

[10] B. M. Gayathri, c. P. Sumathi, phd, an automated technique using gaussian naïve bayes classifier to classify breast cancer, international journal of computer applications, august 2016

[11] Bokde, n.; cortés, g.a.; álvarez, f.m.; kulat, k.: introduction to r package for pattern sequence based forecasting algorithm. R journal 2017

[12] Bhattacharya, g.; ghosh, k.; chowdhury, a.s. granger causality driven ahp for feature weighted KNN. Pattern recognit. 2017.

[13] Nie, c.-x.; song, f.-t. Analyzing the stock market based on the structure of KNN network. Chaos solitons fractals 2018.

[14] Jasmina đ. Novakovic, alempije veljovic,sinisa s. Ilic,milospapic , experimental study of using the k-nearest neighbour classifier with filter methods , 2018

[15] Si-bao chen, yu-lan xu, chris h. Q. Ding, and bin luo. 2018. A nonnegative locally linear KNN Approach for image recognition. (2018)

[16] Ming zong, xiaofeng zhu, and ruili wang, efficient KNN classification with different numbers of nearest neighbors shichao zhang, senior member, ieee, xuelong li, fellow, ieee 2018

[17] Himani sharma,sunil kumar, a survey on decision tree algorithms of classification in data mining, april 2016,international journal of science and research (ijsr]

[18] Mohamed ahmed, ahmet rizaner, ali hakan ulusoy, a novel decision tree classification based on post-pruning with bayes minimum risk ahmed, yong deng, southwest university, china, 2018.

[19] junhao wen , 2 and cheng zhang , an improved random forest algorithm for predicting employee turnover xiang gao , hindawi mathematical problems in engineering , 2019.

[20] te han, dongxiang jiang comparison of random forest, artificial neural networks and support vector Machinefor intelligent diagnosis of rotating machineery, may transactions of the institute of measurement and control, 2018.

[21] Arnu pretorius∗, surette bierman and sarel j. Steel, a meta-analysis of research in random forests for classification, pattern recognition association of south africa and robotics and mechatronics international conference (prasa-robmech) stellenbosch, south africa, 2016.

[22] Raphael couronné , philipp probst and anne-laure boulesteix , random forest versus logistic regression: a large-scale benchmark experiment , couronné et al. Bmc bioinformatics (2018).

[23] A.-a. Nahid and y. Kong, "involvement of Machine learning for breast cancer image classification: a survey," computational and mathematical methods in medicine, pp. 1-29, 2017.

[24] A jasper huang , fabio di troia , and mark stamp, coustic gait analysis using support vector machinee, 2018.

[25]Li yueyang1, a and yi zhi wang1, b , detecting opinion polarities using ensemble of classification algorithms , iop conf. Series: journal of physics: 2019.

[26] Rukshan batuwita∗ and vasile palade† ∗ singapore-mit , class imbalance learning methods for support vector machinees, alliance for research and technology centre;†university of oxford.2016

[27] Qingyao wu, mingkui tan, hengjie song, jian chen, and michael k. Ng, ml-forest: a multi-label tree ensemble method for multi-label classification, ieee transactions on knowledge and data engineering,2016

[28] Yun wan , dr. Qigang gao ,an ensemble sentiment classification system of twitter data for airline service analysis, 2015

[29] Amit gupte, sourabh joshi, pratik gadgul, akshay , comparative study of classification algorithms used in sentiment analysis kadam , ijcsit) international journal of computer science and information technologies 2014.

[30] Kim, K., et al. "General Labelled Data Generator Framework for Network Machine learning." IEEE Xplore, 1 Feb. 2018

[31] Ahmad, Iftikhar, et al. "Performance Comparison of Support Vector Machinee, Random Forest, and Extreme Learning Machinefor Intrusion Detection." IEEE Access, vol. 6, 2018

[32] Naik, Amrita, and Lilavati Samant. "Correlation Review of Classification Algorithm Using Data Mining Tool: WEKA, Rapidminer, Tanagra, Orange and Knime." Procedia Computer Science, vol. 85, 2016

[33] Gislason, Pall Oskar, et al. "Random Forests for Land Cover Classification." Pattern Recognition Letters, vol. 27, no. 4, Mar. 2006

[34] Pondhu, Laxmi Narayana, and Govardhani Kummari. "Performance Analysis of Machine learning Algorithms for Gender Classification." 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018

[35] Maxwell, Aaron E., et al. "Implementation of Machinee-Learning Classification in Remote Sensing: An Applied Review." International Journal of Remote Sensing, vol. 39, no. 9, 2 Feb. 2018

[36] Roadknight, C., et al. "Teaching Key Machine learning Principles Using Anti-Learning Datasets." IEEE Xplore, 1 Dec. 2018

[37] HUBEL, D H, and T N WIESEL. "Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex." The Journal of Physiology, vol. 160, no. 1, 1962

[38] Fukushima, Kunihiko. “Neocognitron: A Self-Organizing Neural Network Approach for a Mechanism of Pattern Recognition Unaffected by Shift in Position.” Biological Cybernetics, vol. 36, no. 4, Apr. 1980

[39] Bülbül, H. İ, et al. “Analysis for Status of the Road Accident Occurance and Determination of the Risk of Accident by Machine learning in Istanbul.” IEEE Xplore, 1 Dec. 2016

[40] https://www.sciencedirect.com/

Appendix :

Experiments Using Hybrid Approach :

asim-alvi (1).ipynb

File Edit View Insert Runtime Tools Help Last saved at 5:31 PM

Comment Share

+ Code + Text

Connect Editing

```
# libraries

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import matplotlib.pyplot as plt
np.random.seed(32)


from sklearn.model_selection import train_test_split
from sklearn.metrics import roc_auc_score
from sklearn.manifold import TSNE

from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.layers import LSTM, Conv1D, MaxPooling1D, Dropout
from keras.utils.np_utils import to_categorical


%matplotlib inline
```

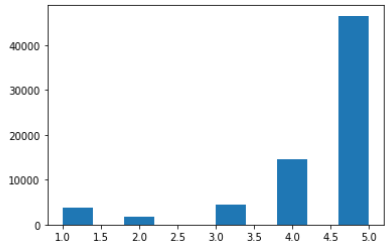
```
df.head()
```

	id	brand	categories	dateAdded	dateUpdated	ean	keys	manufacturer	manufact
0	AV13O1A8GV-KLJ3akUyj	Universal Music	Movies, Music & Books,Music,R&b,Movies & TV,Mo...	2017-07-25T00:52:42Z	2018-02-05T08:36:58Z	6.02537E+11	602537205981,universalmusic/14331328,universal...	Universal Music Group / Cash Money	
1	AV14LG0R-jtxr-f38QfS	Lundberg	Food,Packaged Foods,Snacks,Crackers,Snacks, Co...	2017-07-25T05:16:03Z	2018-02-05T11:27:45Z	73416000391	lundbergorganiccinnamontoastricecakes/b000fvzw...	Lundberg	
2	AV14LG0R-jtxr-f38QfS	Lundberg	Food,Packaged Foods,Snacks,Crackers,Snacks, Co...	2017-07-25T05:16:03Z	2018-02-05T11:27:45Z	73416000391	lundbergorganiccinnamontoastricecakes/b000fvzw...	Lundberg	
3	AV16khLE-jtxr-f38VFn	K-Y	Personal Care,Medicine Cabinet,Lubricant/Sperm...	2017-07-25T16:26:19Z	2018-02-05T11:25:51Z	67981934427	kylovesensualitypleasuregel/b00u2whx8s,0679819...	K-Y	6

5 rows × 25 columns

```
plt.hist(df['reviews.rating'])
```

```
(array([ 3701.,     0.,  1833.,     0.,     0.,  4369.,     0., 14598.,
            0., 46543.]),
 array([1. , 1.4, 1.8, 2.2, 2.6, 3. , 3.4, 3.8, 4.2, 4.6, 5. ]),
 <BarContainer object of 10 artists>)
```



```
[ ] train_text, test_text, train_y, test_y = train_test_split(df['reviews.text'],df['target'],test_size = 0.2)
```

```
[ ] train_text.shape
```

```
(56835,)
```

```
MAX_NB_WORDS = 20000

# get the raw text data
texts_train = train_text.astype(str)
texts_test = test_text.astype(str)

# finally, vectorize the text samples into a 2D integer tensor
tokenizer = Tokenizer(nb_words=MAX_NB_WORDS, char_level=False)
tokenizer.fit_on_texts(texts_train)
sequences = tokenizer.texts_to_sequences(texts_train)
sequences_test = tokenizer.texts_to_sequences(texts_test)

word_index = tokenizer.word_index
print('Found %s unique tokens.' % len(word_index))
```

```
767]
```

```
[ ] type(tokenizer.word_index), len(tokenizer.word_index)
```

```
(dict, 26057)
```

```
[ ] index_to_word = dict((i, w) for w, i in tokenizer.word_index.items())
```

```
" ".join([index_to_word[i] for i in sequences[0]])
```

```
"i'd like the old tide back it cleaned so well and left clothes smelling great but the addition of acti lift has changed everything clothes come out stiff rather
```

```
[ ] seq_lens = [len(s) for s in sequences]
print("average length: %0.1f" % np.mean(seq_lens))
print("max length: %d" % max(seq_lens))
```

```
average length: 39.4
max length: 1034
```

asim-alvi (1).ipynb

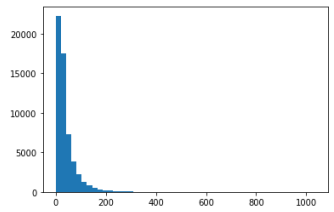
File Edit View Insert Runtime Tools Help Last saved at 5:31 PM

Comment Share

+ Code + Text

Connect Editing

```
%matplotlib inline
import matplotlib.pyplot as plt

plt.hist(seq_lens, bins=50);
```




```
[ ] plt.hist([l for l in seq_lens if l < 200], bins=50);
```

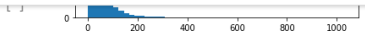
asim-alvi (1).ipynb

File Edit View Insert Runtime Tools Help Last saved at 5:31 PM

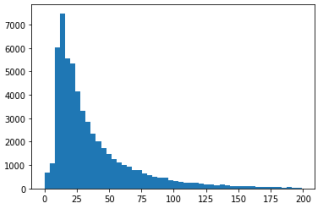
Comment Share

+ Code + Text

Connect Editing




```
plt.hist([l for l in seq_lens if l < 200], bins=50);
```




```
[ ] MAX_SEQUENCE_LENGTH = 150

    # pad sequences with 0s
    x_train = pad_sequences(sequences, maxlen=MAX_SEQUENCE_LENGTH)
    x_test = pad_sequences(sequences_test, maxlen=MAX_SEQUENCE_LENGTH)
    print('Shape of data tensor:', x_train.shape)
```

```
[ ] MAX_SEQUENCE_LENGTH = 150

    # pad sequences with 0s
    x_train = pad_sequences(sequences, maxlen=MAX_SEQUENCE_LENGTH)
    x_test = pad_sequences(sequences_test, maxlen=MAX_SEQUENCE_LENGTH)
    print('Shape of data tensor:', x_train.shape)
    print('Shape of data test tensor:', x_test.shape)
```

```
Shape of data tensor: (56835, 150)
Shape of data test tensor: (14209, 150)
```

```
y_train = train_y
y_test = test_y

y_train = to_categorical(np.asarray(y_train))
print('Shape of label tensor:', y_train.shape)
```

```
Shape of label tensor: (56835, 2)
```

```
from keras.layers import Dense, Input, Flatten
from keras.layers import GlobalAveragePooling1D, Embedding
from keras.models import Model

EMBEDDING_DIM = 50
N_CLASSES = 2
```


```
[ ] MAX_SEQUENCE_LENGTH = 150

    # pad sequences with 0s
    x_train = pad_sequences(sequences, maxlen=MAX_SEQUENCE_LENGTH)
    x_test = pad_sequences(sequences_test, maxlen=MAX_SEQUENCE_LENGTH)
    print('Shape of data tensor:', x_train.shape)
    print('Shape of data test tensor:', x_test.shape)
```

```
Shape of data tensor: (56835, 150)
Shape of data test tensor: (14209, 150)
```

```
y_train = train_y
y_test = test_y

y_train = to_categorical(np.asarray(y_train))
print('Shape of label tensor:', y_train.shape)
```

```
Shape of label tensor: (56835, 2)
```

```
from keras.layers import Dense, Input, Flatten
from keras.layers import GlobalAveragePooling1D, Embedding
from keras.models import Model

EMBEDDING_DIM = 50
N_CLASSES = 2
```

```
from keras.layers import Dense, Input, Flatten
from keras.layers import GlobalAveragePooling1D, Embedding
from keras.models import Model

EMBEDDING_DIM = 50
N_CLASSES = 2

# input: a sequence of MAX_SEQUENCE_LENGTH integers
sequence_input = Input(shape=(MAX_SEQUENCE_LENGTH,), dtype='int32')

embedding_layer = Embedding(MAX_NB_WORDS, EMBEDDING_DIM,
                            input_length=MAX_SEQUENCE_LENGTH,
                            trainable=True)
embedded_sequences = embedding_layer(sequence_input)

average = GlobalAveragePooling1D()(embedded_sequences)
predictions = Dense(N_CLASSES, activation='softmax')(average)

model = Model(sequence_input, predictions)
model.compile(loss='categorical_crossentropy',
              optimizer='adam', metrics=['acc'])
```

asim-alvi (1).ipynb

File Edit View Insert Runtime Tools Help Last saved at 5:31 PM

+ Code + Text

```
model.fit(x_train, y_train, validation_split=0.1,
          epochs=10, batch_size=128)
```

```
Epoch 1/10
400/400 [==============================] - 13s 33ms/step - loss: 0.4019 - acc: 0.8573 - val_loss: 0.3583 - val_acc: 0.8633
Epoch 2/10
400/400 [==============================] - 13s 31ms/step - loss: 0.3216 - acc: 0.8719 - val_loss: 0.3041 - val_acc: 0.8777
Epoch 3/10
400/400 [==============================] - 13s 32ms/step - loss: 0.2696 - acc: 0.8895 - val_loss: 0.2697 - val_acc: 0.8886
Epoch 4/10
400/400 [==============================] - 13s 32ms/step - loss: 0.2384 - acc: 0.9030 - val_loss: 0.2523 - val_acc: 0.8995
Epoch 5/10
400/400 [==============================] - 13s 32ms/step - loss: 0.2199 - acc: 0.9118 - val_loss: 0.2444 - val_acc: 0.9043
Epoch 6/10
400/400 [==============================] - 12s 30ms/step - loss: 0.2068 - acc: 0.9177 - val_loss: 0.2385 - val_acc: 0.9071
Epoch 7/10
400/400 [==============================] - 12s 30ms/step - loss: 0.1965 - acc: 0.9231 - val_loss: 0.2360 - val_acc: 0.9094
Epoch 8/10
400/400 [==============================] - 12s 30ms/step - loss: 0.1882 - acc: 0.9273 - val_loss: 0.2374 - val_acc: 0.9085
Epoch 9/10
400/400 [==============================] - 12s 30ms/step - loss: 0.1810 - acc: 0.9305 - val_loss: 0.2327 - val_acc: 0.9115
Epoch 10/10
400/400 [==============================] - 12s 30ms/step - loss: 0.1745 - acc: 0.9335 - val_loss: 0.2327 - val_acc: 0.9133
<tensorflow.python.keras.callbacks.History at 0x29fb87d36d8>
```

```
[ ] output_test = model.predict(x_test)
    print("test auc:", roc_auc_score(y_test,output_test[:,1]))
```

```
test auc: 0.9121867206388242
```

Experiment on another Dataset

```
# Support vector classifier
from sklearn.svm import SVC
svc_classifier=SVC()
svc_classifier.fit(X_train,y_train)
y_pred_scv=svc_classifier.predict(X_test)
y_pred_scv=svc_classifier.predict(X_test)
accuracy_score(y_test,y_pred_scv)
```

```
packages\sklearn\svm\base.py:193: FutureWarning: The default value of gamma wil
```

```
[ ]
```

```
[ ] # Train with standard scaled data
    svc_classifier2=SVC()
    svc_classifier2.fit(X_train_sc,y_train)
    y_pred_svc_sc=svc_classifier2.predict(X_test_sc)
    accuracy_score(y_test,y_pred_svc_sc)
```

```
0.9649122807017544
```

```
[ ] # Logistic Regression
    from sklearn.linear_model import LogisticRegression
    lr_classifier= LogisticRegression(random_state=51,penalty='l1')
    lr_classifier.fit(X_train,y_train)
    y_pred_lr=lr_classifier.predict(X_test)
    y_pred_lr=lr_classifier.predict(X_test)
    accuracy_score(y_test,y_pred_lr)
```

```
c:\users\patid\appdata\local\programs\python\python37\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWa
  FutureWarning)
c:\users\patid\appdata\local\programs\python\python37\lib\site-packages\sklearn\svm\base.py:929: ConvergenceWarning: l
  "the number of iterations.", ConvergenceWarning)
0.9736842105263158
```

```
# Train with standard scaled Data
lr_classifier2=LogisticRegression(random_state=51,penalty='l1')
lr_classifier2.fit(X_train_sc,y_train)
y_pred_lr=lr_classifier.predict(X_test)
accuracy_score(y_test,y_pred_lr)
```

```
c:\users\patid\appdata\local\programs\python\python37\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWa
  FutureWarning)
0.9736842105263158
```

```
[ ] # K- Nearest Neighbour Classifier
    from sklearn.neighbors import KNeighborsClassifier
    knn_classifier=KNeighborsClassifier(n_neighbors=5,metric='minkowski',p=2)
    knn_classifier.fit(X_train,y_train)
    y_pred_knn=knn_classifier.predict(X_test)
    accuracy_score(y_test,y_pred_knn)
```

```
0.9385964912280702
```

```
# Train with Standard scaled Data
knn_classifier2=KNeighborsClassifier(n_neighbors=5,metric='minkowski',p=2)
knn_classifier2.fit(X_train_sc,y_train)
y_pred_knn_sc=knn_classifier.predict(X_test_sc)
accuracy_score(y_test,y_pred_knn_sc)
```

```
0.5789473684210527
```

```
# Naive bayes Classifier
from sklearn.naive_bayes import GaussianNB
nb_classifier = GaussianNB()
nb_classifier.fit(X_train, y_train)
y_pred_nb = nb_classifier.predict(X_test)
accuracy_score(y_test, y_pred_nb)
```

```
0.9473684210526315
```

```
[ ] # Train with Standard scaled Data
    nb_classifier2 = GaussianNB()
    nb_classifier2.fit(X_train_sc, y_train)
    y_pred_nb_sc = nb_classifier2.predict(X_test_sc)
    accuracy_score(y_test, y_pred_nb_sc)
```

```
0.9385964912280702
```

```
[ ] # Decision Tree Classifier
    from sklearn.tree import DecisionTreeClassifier
    dt_classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 51)
    dt_classifier.fit(X_train, y_train)
    y_pred_dt = dt_classifier.predict(X_test)
    accuracy_score(y_test, y_pred_dt)
```

```
0.9473684210526315
```

```
[ ] # Train with Standard scaled Data
    dt_classifier2=DecisionTreeClassifier(criterion='entropy',random_state=51)
    dt_classifier2.fit(X_train_sc,y_train)
    y_pred_sc=dt_classifier.predict(X_test_sc)
    accuracy_score(y_test,y_pred_sc)
```

```
0.7543859649122807
```

```
[ ] # Random Forest Classifier
    from sklearn.ensemble import RandomForestClassifier
    rf_classifier=RandomForestClassifier(n_estimators=20,criterion='entropy',random_state=51)
    rf_classifier.fit(X_train,y_train)
    y_pred=rf_classifier.predict(X_test)
    accuracy_score(y_test,y_pred)
```

```
0.9736842105263158
```

```
rf_classifier2=RandomForestClassifier(n_estimators=20,criterion='entropy',random_state=51)
rf_classifier2.fit(X_train_sc,y_train)
y_pred_sc=rf_classifier2.predict(X_test_sc)
accuracy_score(y_test,y_pred_sc)
```

```
0.9736842105263158
```

```
# Adaboost Classifier
from sklearn.ensemble import AdaBoostClassifier
adb_classifier=AdaBoostClassifier(DecisionTreeClassifier(criterion='entropy',random_state=200),
                                  n_estimators=2000,
                                  learning_rate=0.1,
                                  algorithm='SAMME.R',
                                  random_state=1,)
adb_classifier.fit(X_train,y_train)
y_pred_adb=adb_classifier.predict(X_test)
accuracy_score(y_test,y_pred_adb)
```

```
0.9473684210526315
```

```
[ ] # Train wtih scaled Data
    adb_classifier2=AdaBoostClassifier(DecisionTreeClassifier(criterion='entropy',random_state=200),
                                       n_estimators=2000,
                                       learning_rate=0.1,
                                       algorithm='SAMME.R',
                                       random_state=1,)
    adb_classifier2.fit(X_train_sc,y_train)
    y_pred_adb_sc=adb_classifier2.predict(X_test_sc)
    accuracy_score(y_test,y_pred_adb_sc)
```

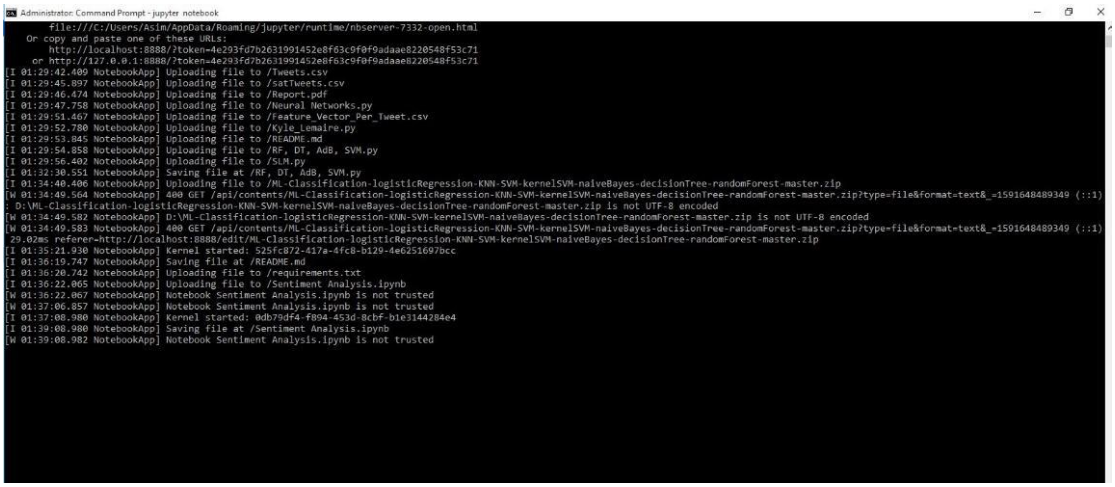
```
0.9473684210526315
```

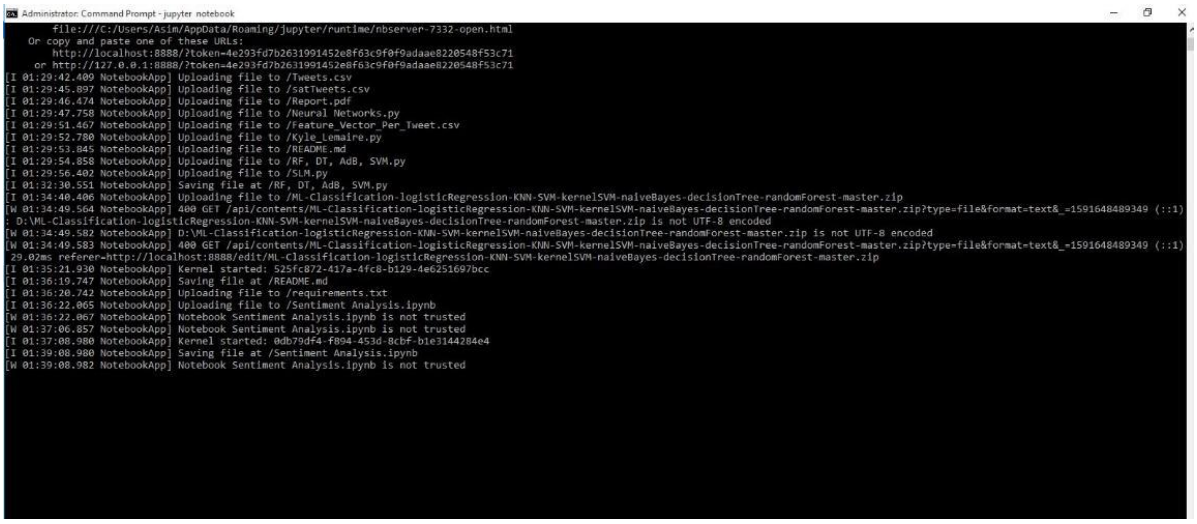
```
# XGBoost classifier
from xgboost import XGBClassifier
xgb_classifier=XGBClassifier()
xgb_classifier.fit(X_train,y_train)
y_pred_xgb=xgb_classifier.predict(X_test)
accuracy_score(y_test,y_pred_xgb)
```

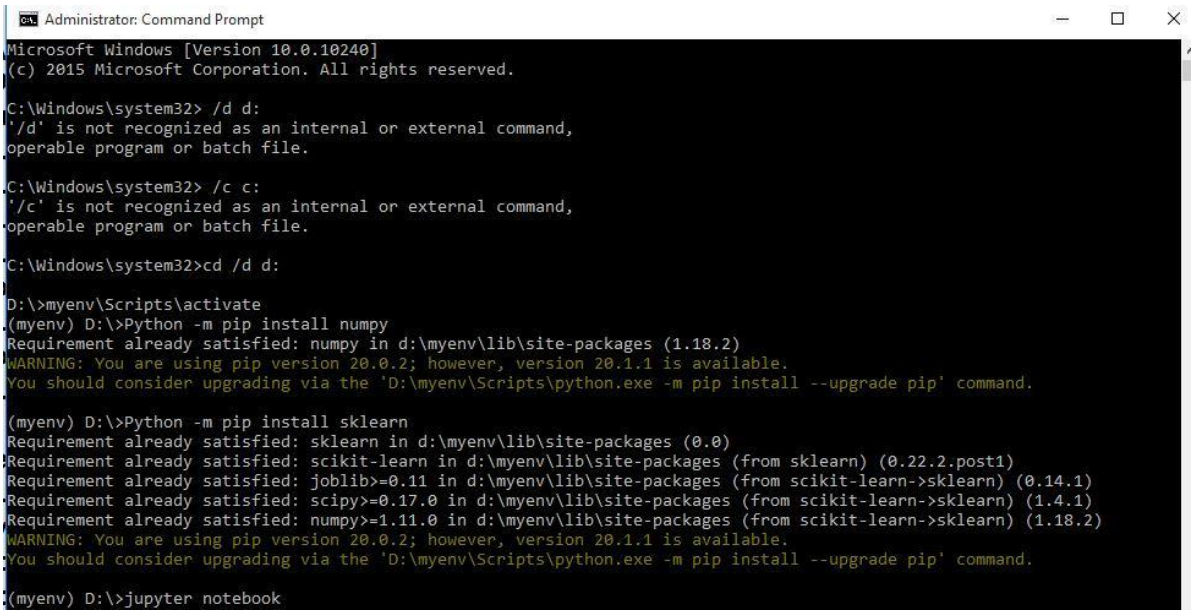
```
0.9824561403508771
```

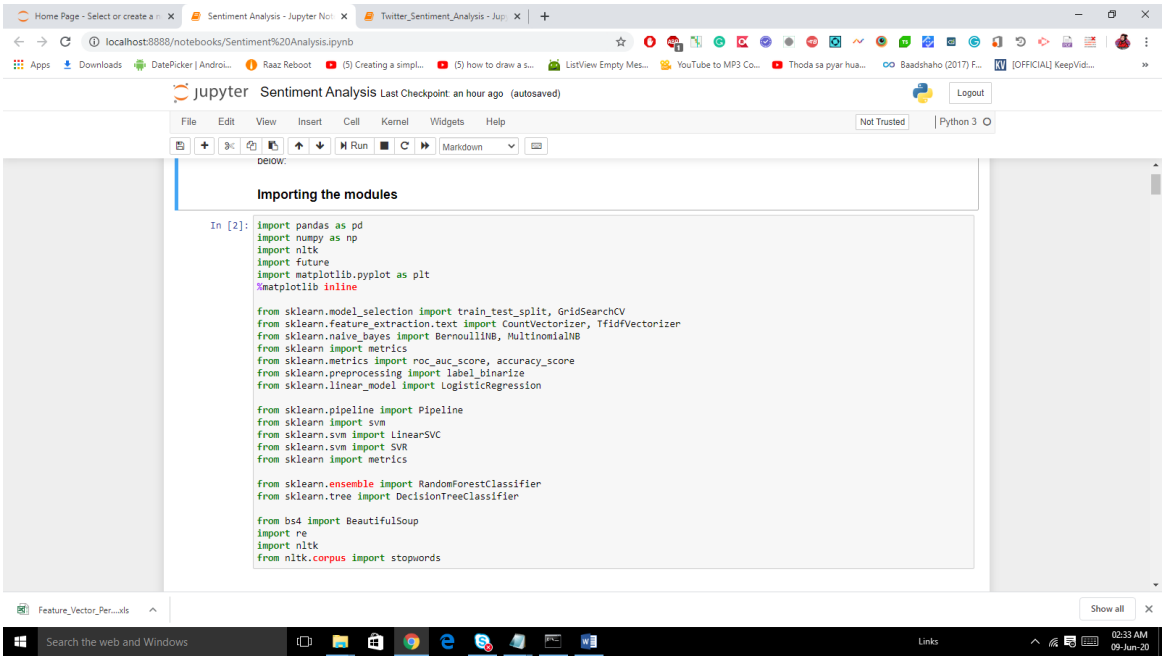
```
[ ] # Train with Standard Sclaed Data
    xgb_classifier2=XGBClassifier()
    xgb_classifier2.fit(X_train_sc,y_train)
    y_pred_xgb_sc=xgb_classifier2.predict(X_test_sc)
    accuracy_score(y_test,y_pred_xgb_sc)
```

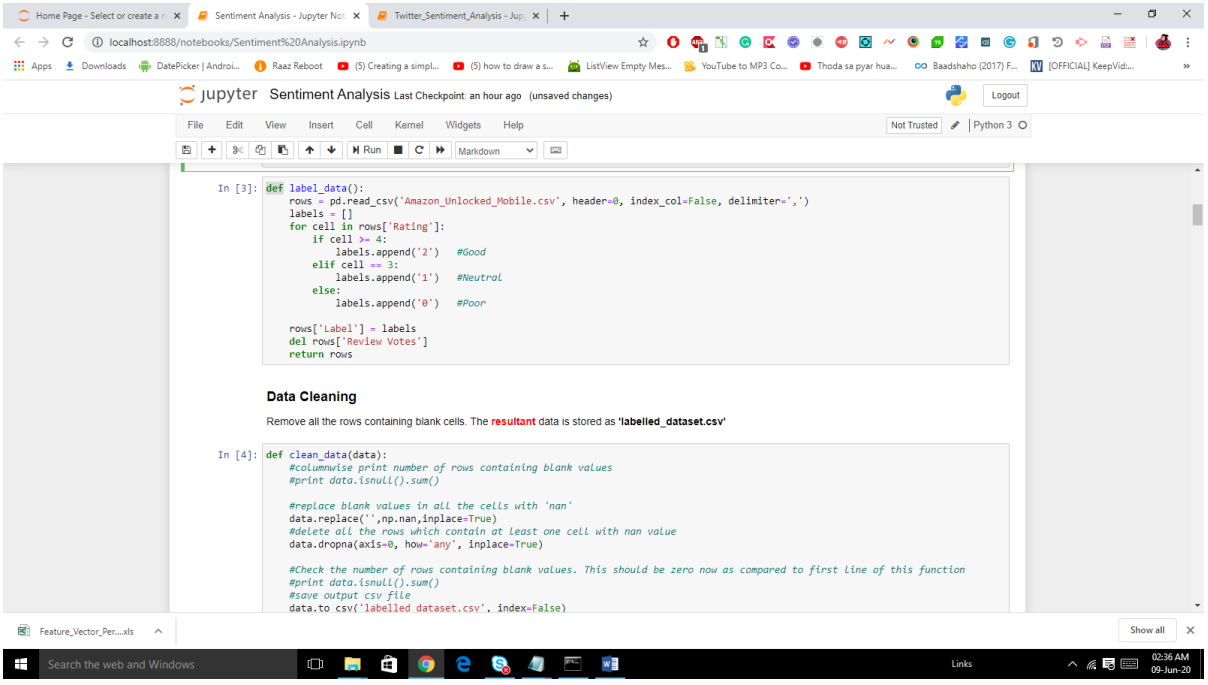
```
0.9824561403508771
```

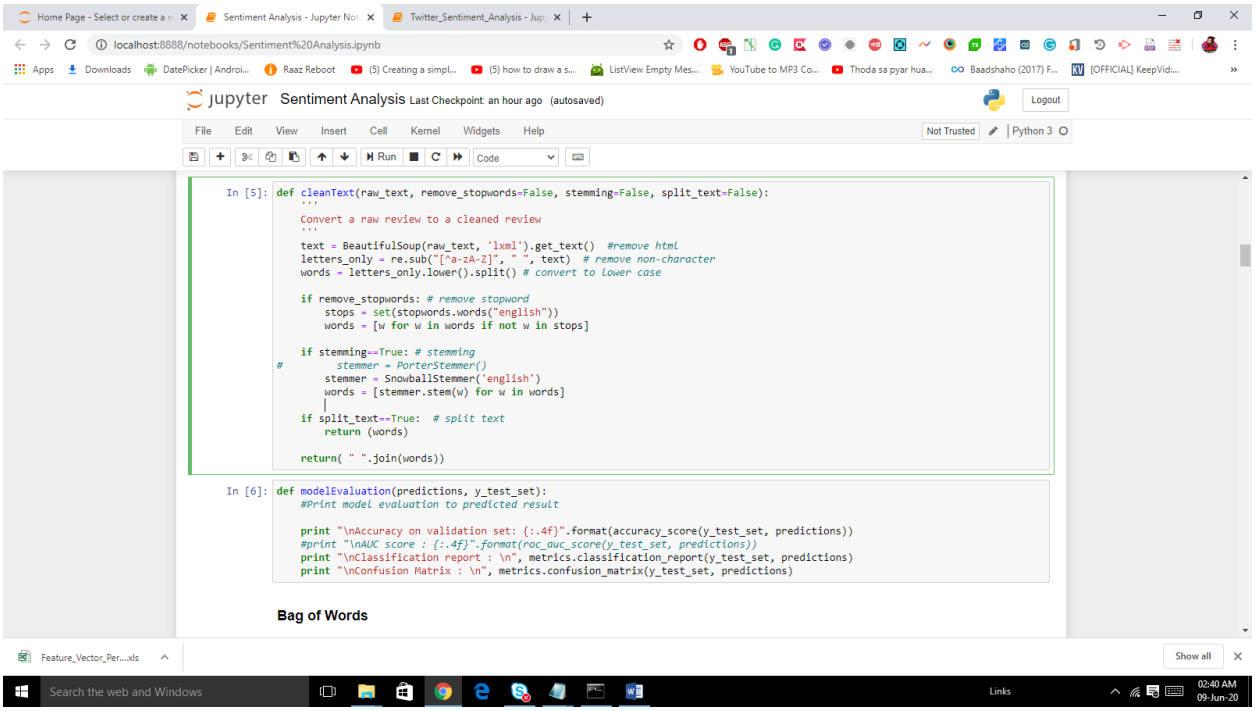

Administrator: Command Prompt - jupyter notebook
file:///C:/Users/Asim/AppData/Roaming/jupyter/runtime/nbserver-7332-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=4e293fd7b2631991452e8f63c9f0f9adaae8220548f53c71
or http://127.0.0.1:8888/?token=4e293fd7b2631991452e8f63c9f0f9adaae8220548f53c71
[I 01:29:42.409 NotebookApp] Uploading file to /Tweets.csv
[I 01:29:45.897 NotebookApp] Uploading file to /satTweets.csv
[I 01:29:46.474 NotebookApp] Uploading file to /Report.pdf
[I 01:29:47.758 NotebookApp] Uploading file to /Neural Networks.py
[I 01:29:51.467 NotebookApp] Uploading file to /Feature_Vector_Per_Tweet.csv
[I 01:29:52.780 NotebookApp] Uploading file to /Kyle_Lemaire.py
[I 01:29:53.845 NotebookApp] Uploading file to /README.md
[I 01:29:54.858 NotebookApp] Uploading file to /RF, DT, AdB, SVM.py
[I 01:29:56.402 NotebookApp] Uploading file to /SLM.py
[I 01:32:30.551 NotebookApp] Saving file at /RF, DT, AdB, SVM.py
[I 01:34:40.406 NotebookApp] Uploading file to /ML-Classification-logisticRegression-KNN-SVM-kernelSVM-naiveBayes-decisionTree-randomForest-master.zip
[W 01:34:49.564 NotebookApp] 400 GET /api/contents/ML-Classification-logisticRegression-KNN-SVM-kernelSVM-naiveBayes-decisionTree-randomForest-master.zip?type=file&format=text&_=1591648489349 (::1)
: D:\ML-Classification-logisticRegression-KNN-SVM-kernelSVM-naiveBayes-decisionTree-randomForest-master.zip is not UTF-8 encoded
[W 01:34:49.582 NotebookApp] D:\ML-Classification-logisticRegression-KNN-SVM-kernelSVM-naiveBayes-decisionTree-randomForest-master.zip is not UTF-8 encoded
[W 01:34:49.583 NotebookApp] 400 GET /api/contents/ML-Classification-logisticRegression-KNN-SVM-kernelSVM-naiveBayes-decisionTree-randomForest-master.zip?type=file&format=text&_=1591648489349 (::1)
29.02ms referer=http://localhost:8888/edit/ML-Classification-logisticRegression-KNN-SVM-kernelSVM-naiveBayes-decisionTree-randomForest-master.zip
[I 01:35:21.930 NotebookApp] Kernel started: 525fc872-417a-4fc8-b129-4e6251697bcc
[I 01:36:19.747 NotebookApp] Saving file at /README.md
[I 01:36:20.742 NotebookApp] Uploading file to /requirements.txt
[I 01:36:22.065 NotebookApp] Uploading file to /Sentiment Analysis.ipynb
[W 01:36:22.067 NotebookApp] Notebook Sentiment Analysis.ipynb is not trusted
[W 01:37:06.857 NotebookApp] Notebook Sentiment Analysis.ipynb is not trusted
[I 01:37:08.980 NotebookApp] Kernel started: 0db79df4-f894-453d-8cbf-b1e3144284e4
[I 01:39:08.980 NotebookApp] Saving file at /Sentiment Analysis.ipynb
[W 01:39:08.982 NotebookApp] Notebook Sentiment Analysis.ipynb is not trusted


Administrator: Command Prompt - jupyter notebook
file:///C:/Users/Asim/AppData/Roaming/jupyter/runtime/nbserver-7332-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=4e293fd7b2631991452e8f63c9f0f9adaae8220548f53c71
or http://127.0.0.1:8888/?token=4e293fd7b2631991452e8f63c9f0f9adaae8220548f53c71
[I 01:29:42.409 NotebookApp] Uploading file to /Tweets.csv
[I 01:29:45.897 NotebookApp] Uploading file to /satTweets.csv
[I 01:29:46.474 NotebookApp] Uploading file to /Report.pdf
[I 01:29:47.758 NotebookApp] Uploading file to /Neural Networks.py
[I 01:29:51.467 NotebookApp] Uploading file to /Feature_Vector_Per_Tweet.csv
[I 01:29:52.780 NotebookApp] Uploading file to /Kyle_Lemaire.py
[I 01:29:53.845 NotebookApp] Uploading file to /README.md
[I 01:29:54.858 NotebookApp] Uploading file to /RF, DT, AdB, SVM.py
[I 01:29:56.402 NotebookApp] Uploading file to /SLM.py
[I 01:32:30.551 NotebookApp] Saving file at /RF, DT, AdB, SVM.py
[I 01:34:40.406 NotebookApp] Uploading file to /ML-Classification-logisticRegression-KNN-SVM-kernelSVM-naiveBayes-decisionTree-randomForest-master.zip
[W 01:34:49.564 NotebookApp] 400 GET /api/contents/ML-Classification-logisticRegression-KNN-SVM-kernelSVM-naiveBayes-decisionTree-randomForest-master.zip?type=file&format=text&_=1591648489349 (::1)
: D:\ML-Classification-logisticRegression-KNN-SVM-kernelSVM-naiveBayes-decisionTree-randomForest-master.zip is not UTF-8 encoded
[W 01:34:49.582 NotebookApp] D:\ML-Classification-logisticRegression-KNN-SVM-kernelSVM-naiveBayes-decisionTree-randomForest-master.zip is not UTF-8 encoded
[W 01:34:49.583 NotebookApp] 400 GET /api/contents/ML-Classification-logisticRegression-KNN-SVM-kernelSVM-naiveBayes-decisionTree-randomForest-master.zip?type=file&format=text&_=1591648489349 (::1)
29.02ms referer=http://localhost:8888/edit/ML-Classification-logisticRegression-KNN-SVM-kernelSVM-naiveBayes-decisionTree-randomForest-master.zip
[I 01:35:21.930 NotebookApp] Kernel started: 525fc872-417a-4fc8-b129-4e6251697bcc
[I 01:36:19.747 NotebookApp] Saving file at /README.md
[I 01:36:20.742 NotebookApp] Uploading file to /requirements.txt
[I 01:36:22.065 NotebookApp] Uploading file to /Sentiment Analysis.ipynb
[W 01:36:22.067 NotebookApp] Notebook Sentiment Analysis.ipynb is not trusted
[W 01:37:06.857 NotebookApp] Notebook Sentiment Analysis.ipynb is not trusted
[I 01:37:08.980 NotebookApp] Kernel started: 0db79df4-f894-453d-8cbf-b1e3144284e4
[I 01:39:08.980 NotebookApp] Saving file at /Sentiment Analysis.ipynb
[W 01:39:08.982 NotebookApp] Notebook Sentiment Analysis.ipynb is not trusted


Administrator: Command Prompt
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.
C:\Windows\system32> /d d:
'/d' is not recognized as an internal or external command,
operable program or batch file.
C:\Windows\system32> /c c:
'/c' is not recognized as an internal or external command,
operable program or batch file.
C:\Windows\system32>cd /d d:
D:\>myenv\Scripts\activate
(myenv) D:\>Python -m pip install numpy
Requirement already satisfied: numpy in d:\myenv\lib\site-packages (1.18.2)
WARNING: You are using pip version 20.0.2; however, version 20.1.1 is available.
You should consider upgrading via the 'D:\myenv\Scripts\python.exe -m pip install --upgrade pip' command.
(myenv) D:\>Python -m pip install sklearn
Requirement already satisfied: sklearn in d:\myenv\lib\site-packages (0.0)
Requirement already satisfied: scikit-learn in d:\myenv\lib\site-packages (from sklearn) (0.22.2.post1)
Requirement already satisfied: joblib>=0.11 in d:\myenv\lib\site-packages (from scikit-learn->sklearn) (0.14.1)
Requirement already satisfied: scipy>=0.17.0 in d:\myenv\lib\site-packages (from scikit-learn->sklearn) (1.4.1)
Requirement already satisfied: numpy>=1.11.0 in d:\myenv\lib\site-packages (from scikit-learn->sklearn) (1.18.2)
WARNING: You are using pip version 20.0.2; however, version 20.1.1 is available.
You should consider upgrading via the 'D:\myenv\Scripts\python.exe -m pip install --upgrade pip' command.
(myenv) D:\>jupyter notebook


Sentiment Analysis - Jupyter Not
localhost:8888/notebooks/Sentiment%20Analysis.ipynb
jupyter Sentiment Analysis Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
below:
Importing the modules
In [2]:
import pandas as pd
import numpy as np
import nltk
import future
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.naive_bayes import BernoulliNB, MultinomialNB
from sklearn import metrics
from sklearn.metrics import roc_auc_score, accuracy_score
from sklearn.preprocessing import label_binarize
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline
from sklearn import svm
from sklearn.svm import LinearSVC
from sklearn.svm import SVR
from sklearn import metrics
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from bs4 import BeautifulSoup
import re
import nltk
from nltk.corpus import stopwords
Feature_Vector_Per....xls
Show all
Search the web and Windows
Links
02:33 AM
09-Jun-20

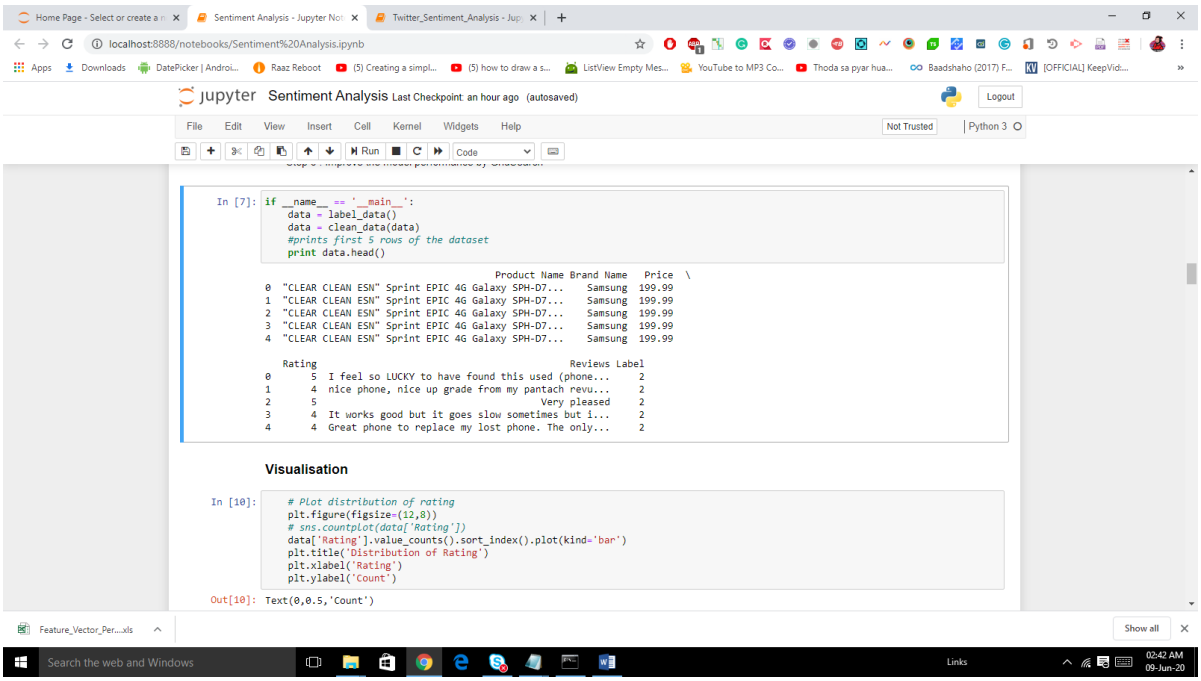

Sentiment Analysis - Jupyter Not
localhost:8888/notebooks/Sentiment%20Analysis.ipynb
jupyter Sentiment Analysis Last Checkpoint: an hour ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
In [3]:
def label_data():
rows = pd.read_csv('Amazon_Unlocked_Mobile.csv', header=0, index_col=False, delimiter=',')
labels = []
for cell in rows['Rating']:
if cell >= 4:
labels.append('2') #Good
elif cell == 3:
labels.append('1') #Neutral
else:
labels.append('0') #Poor
rows['Label'] = labels
del rows['Review Votes']
return rows
Data Cleaning
Remove all the rows containing blank cells. The resultant data is stored as 'labelled_dataset.csv'
In [4]:
def clean_data(data):
#columnwise print number of rows containing blank values
#print data.isnull().sum()
#replace blank values in all the cells with 'nan'
data.replace('',np.nan,inplace=True)
#delete all the rows which contain at least one cell with nan value
data.dropna(axis=0, how='any', inplace=True)
#Check the number of rows containing blank values. This should be zero now as compared to first line of this function
#print data.isnull().sum()
#save output csv file
data.to_csv('labelled_dataset.csv', index=False)
Feature_Vector_Per....xls
Show all
Search the web and Windows
Links
02:36 AM
09-Jun-20



```
In [5]: def cleanText(raw_text, remove_stopwords=False, stemming=False, split_text=False):
            '''
            Convert a raw review to a cleaned review
            '''
            text = BeautifulSoup(raw_text, 'lxml').get_text()  #remove html
            letters_only = re.sub("[^a-zA-Z]", " ", text)  # remove non-character
            words = letters_only.lower().split() # convert to lower case

            if remove_stopwords: # remove stopword
                stops = set(stopwords.words("english"))
                words = [w for w in words if not w in stops]

            if stemming==True: # stemming
        #         stemmer = PorterStemmer()
                stemmer = SnowballStemmer('english')
                words = [stemmer.stem(w) for w in words]

            if split_text==True:  # split text
                return (words)

            return( " ".join(words))
```

```
In [6]: def modelEvaluation(predictions, y_test_set):
            #Print model evaluation to predicted result

            print "\nAccuracy on validation set: {:.4f}".format(accuracy_score(y_test_set, predictions))
            #print "\nAUC score : {:.4f}".format(roc_auc_score(y_test_set, predictions))
            print "\nClassification report : \n", metrics.classification_report(y_test_set, predictions)
            print "\nConfusion Matrix : \n", metrics.confusion_matrix(y_test_set, predictions)
```

Bag of Words

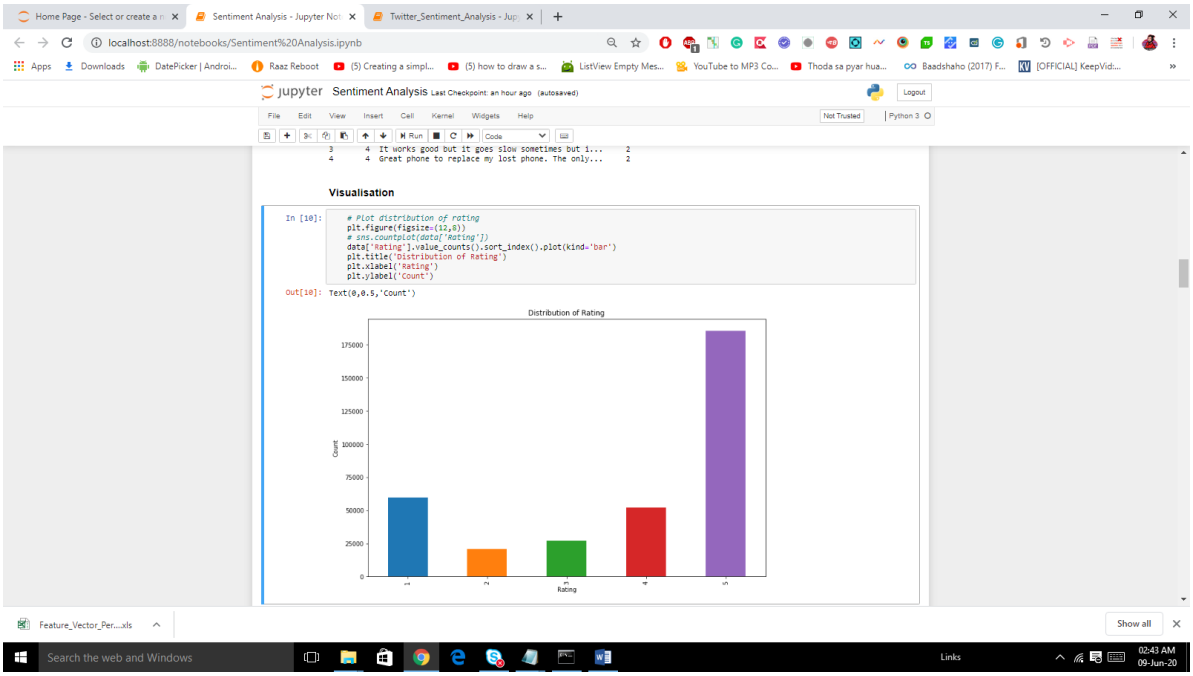


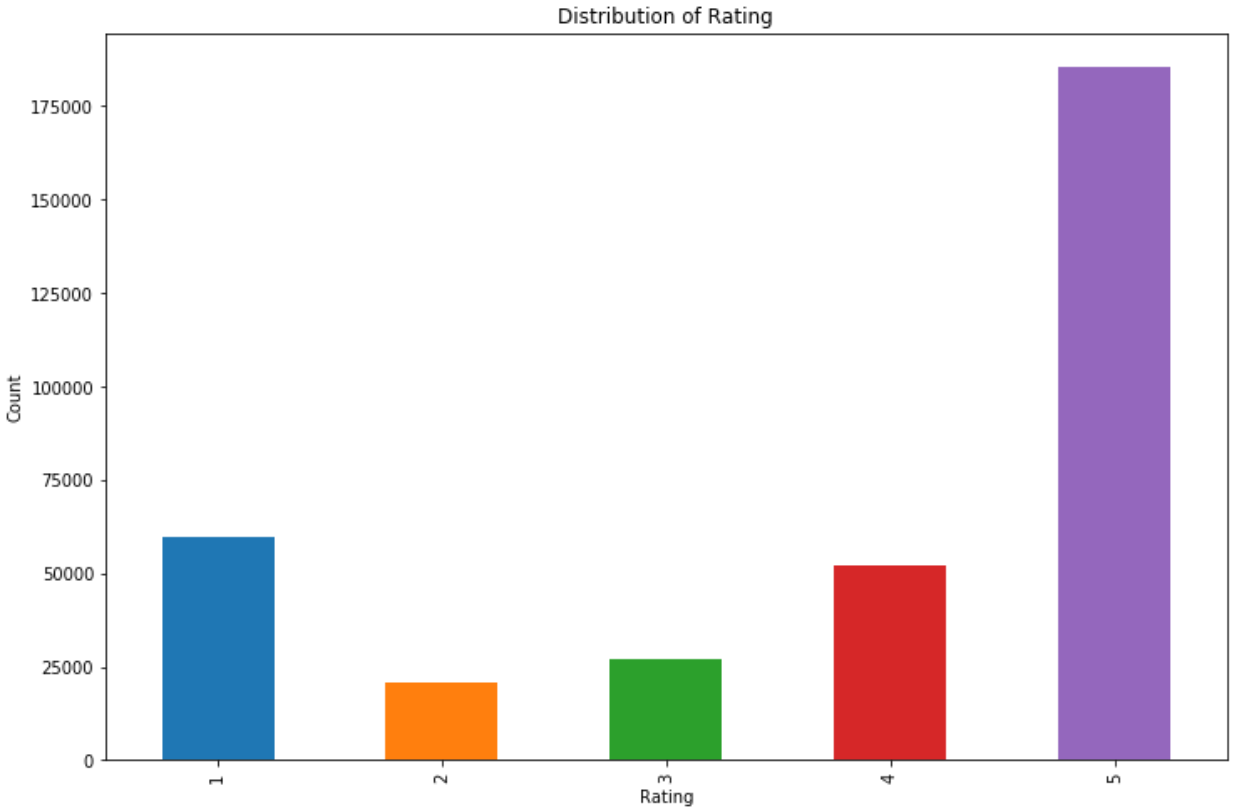
```
In [7]: if __name__ == '__main__':
            data = label_data()
            data = clean_data(data)
            #prints first 5 rows of the dataset
            print data.head()
```

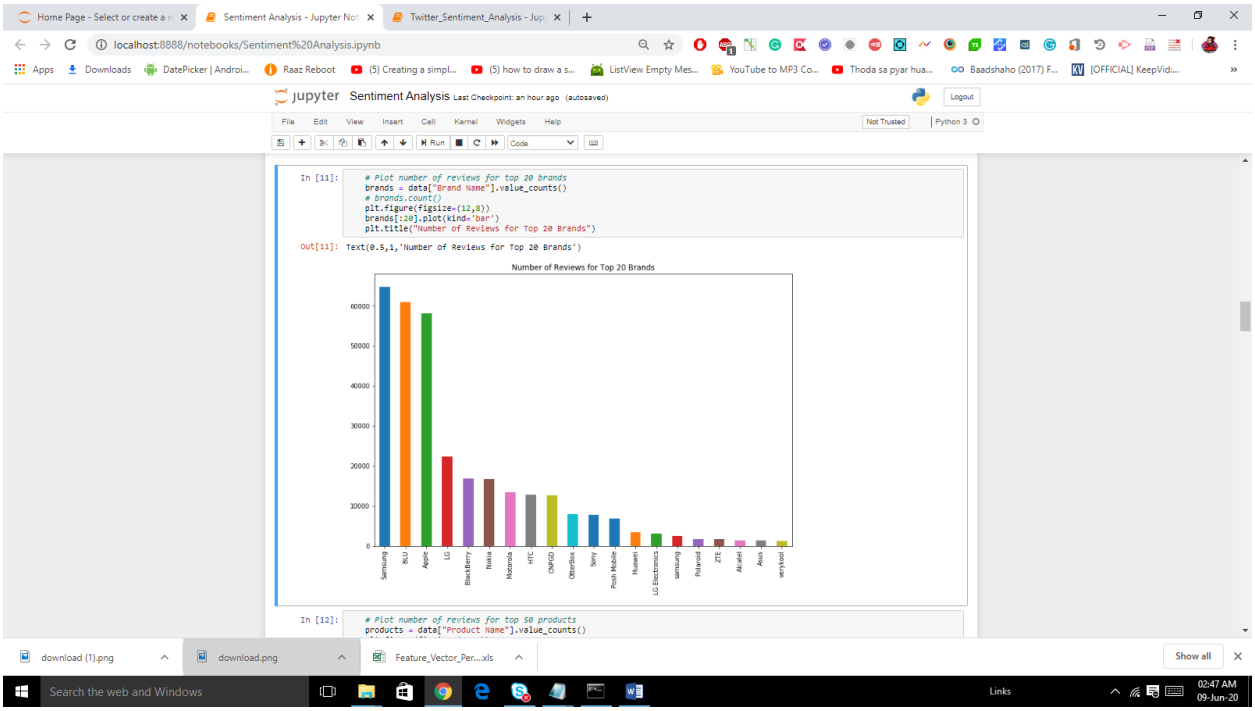
```
                                                Product Name Brand Name   Price  \
0  "CLEAR CLEAN ESN" Sprint EPIC 4G Galaxy SPH-D7...    Samsung  199.99
1  "CLEAR CLEAN ESN" Sprint EPIC 4G Galaxy SPH-D7...    Samsung  199.99
2  "CLEAR CLEAN ESN" Sprint EPIC 4G Galaxy SPH-D7...    Samsung  199.99
3  "CLEAR CLEAN ESN" Sprint EPIC 4G Galaxy SPH-D7...    Samsung  199.99
4  "CLEAR CLEAN ESN" Sprint EPIC 4G Galaxy SPH-D7...    Samsung  199.99

   Rating                                            Reviews Label
0       5  I feel so LUCKY to have found this used (phone...     2
1       4  nice phone, nice up grade from my pantach revu...     2
2       5                                      Very pleased     2
3       4  It works good but it goes slow sometimes but i...     2
4       4  Great phone to replace my lost phone. The only...     2
```

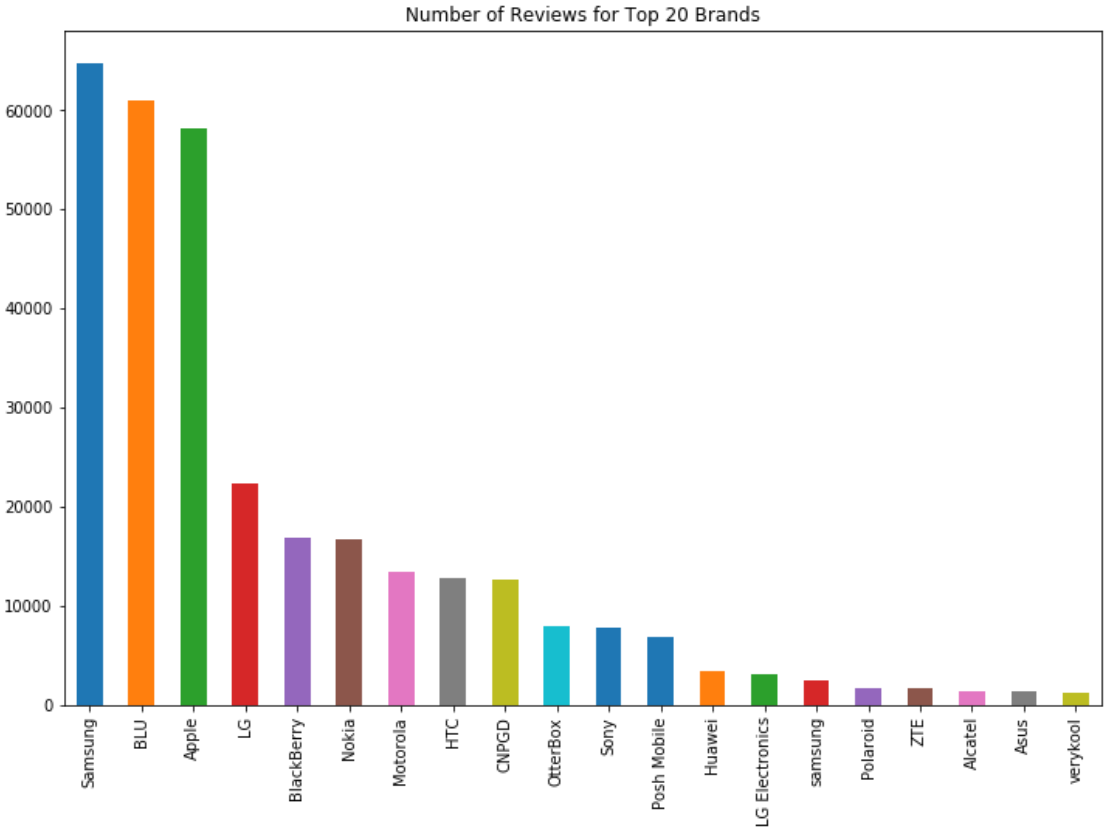
Visualisation

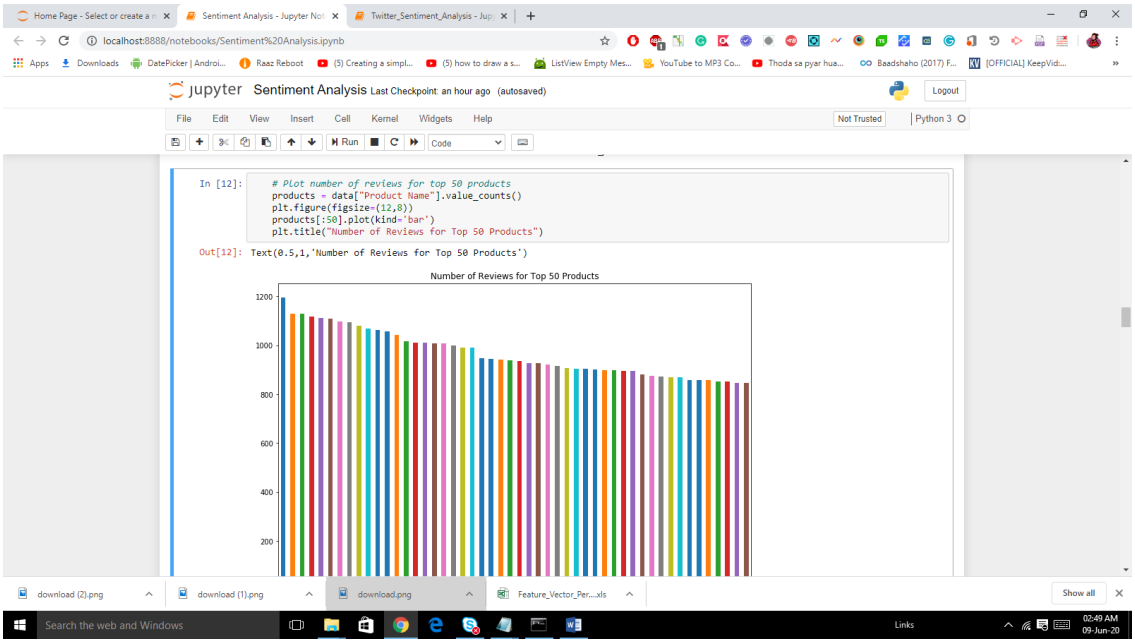
```
In [10]:    # Plot distribution of rating
            plt.figure(figsize=(12,8))
            # sns.countplot(data['Rating'])
            data['Rating'].value_counts().sort_index().plot(kind='bar')
            plt.title('Distribution of Rating')
            plt.xlabel('Rating')
            plt.ylabel('Count')

Out[10]: Text(0,0.5,'Count')
```

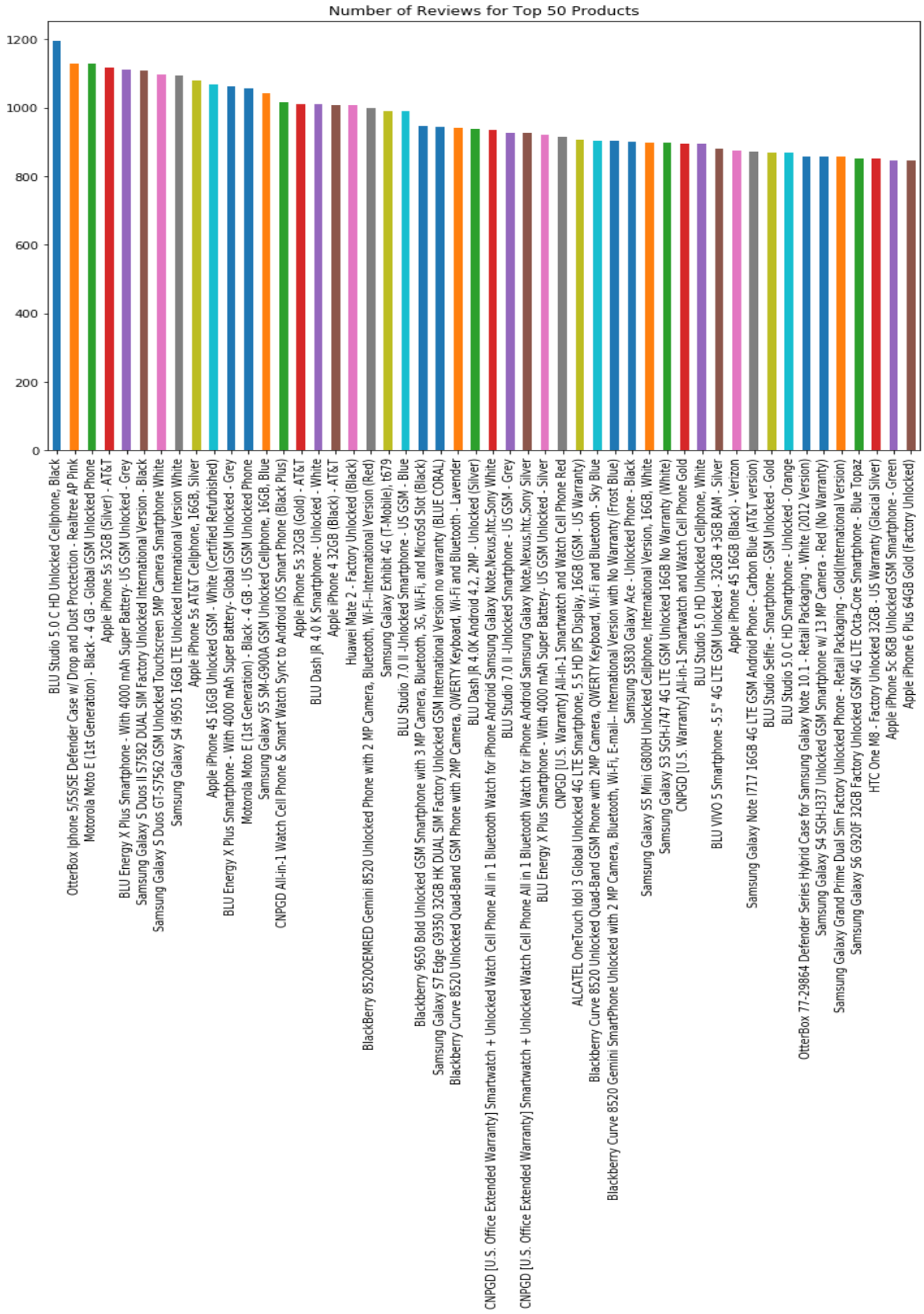

Sentiment Analysis - Jupyter Notebook
localhost:8888/notebooks/Sentiment%20Analysis.ipynb
jupyter Sentiment Analysis
Visualisation
In [10]:
# Plot distribution of rating
plt.figure(figsize=(12,8))
# sns.countplot(data['Rating'])
data['Rating'].value_counts().sort_index().plot(kind='bar')
plt.title('Distribution of Rating')
plt.xlabel('Rating')
plt.ylabel('Count')
Out[10]: Text(0,0.5,'Count')
Distribution of Rating
Count
Rating
175000
150000
125000
100000
75000
50000
25000
0
1
2
3
4
5
Search the web and Windows
Links
02:43 AM
09-Jun-20

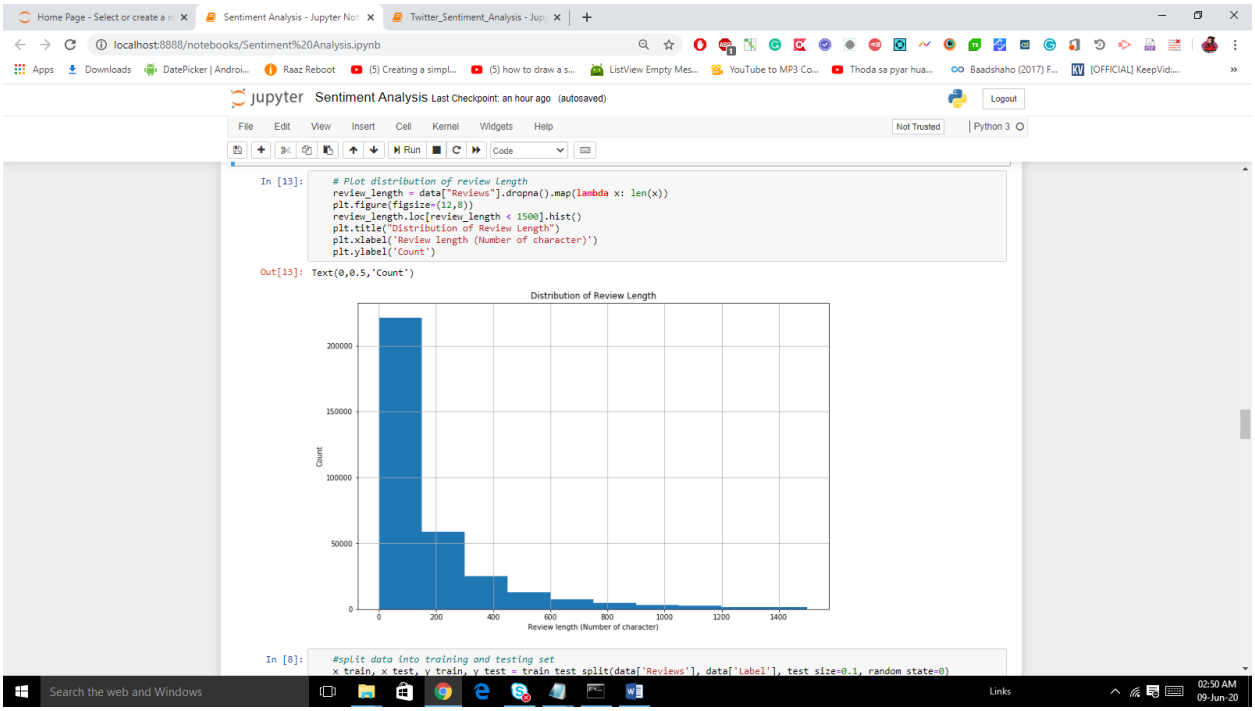

Distribution of Rating
Count
Rating
175000
150000
125000
100000
75000
50000
25000
0
1
2
3
4
5

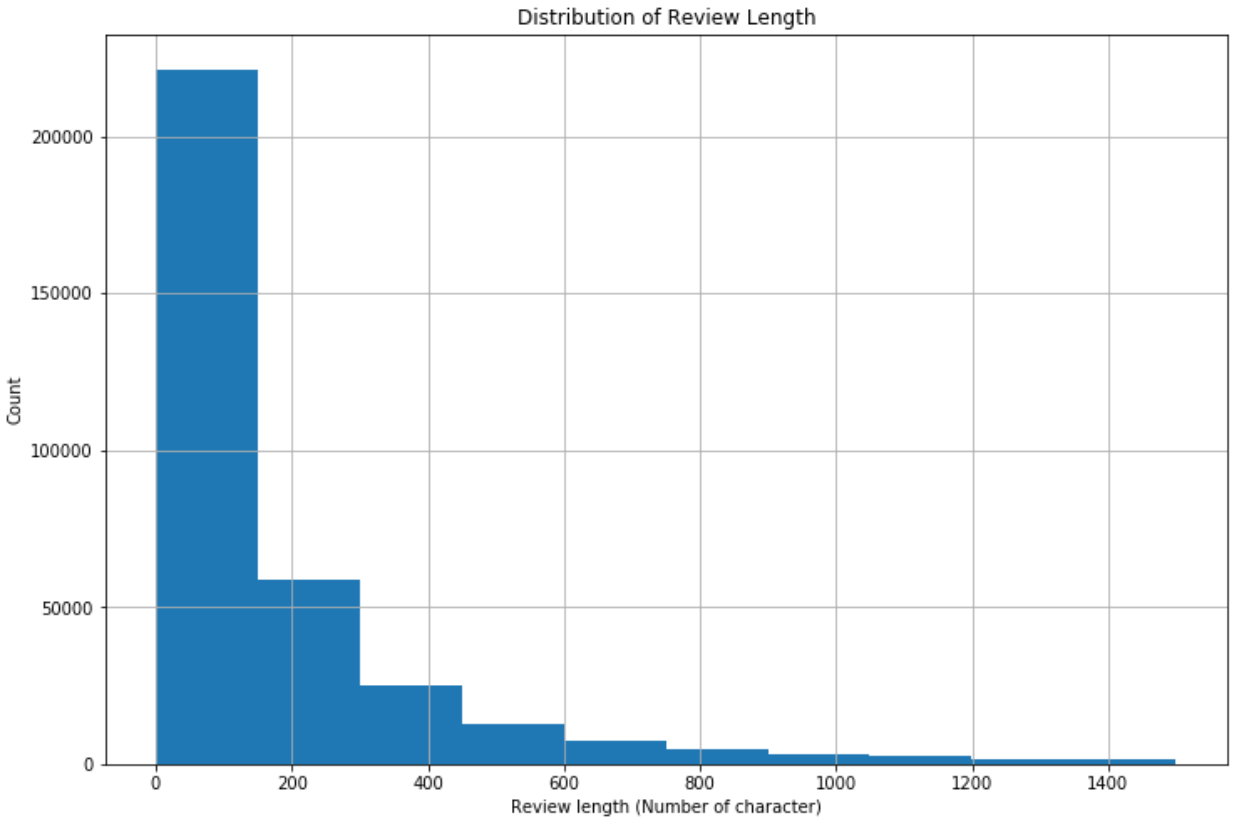

Sentiment Analysis - Jupyter Not
localhost:8888/notebooks/Sentiment%20Analysis.ipynb
jupyter Sentiment Analysis Last Checkpoint: an hour ago (autosaved)
In [11]:
# Plot number of reviews for top 20 brands
brands = data["Brand Name"].value_counts()
# brands.count()
plt.figure(figsize=(12,8))
brands[:20].plot(kind='bar')
plt.title("Number of Reviews for Top 20 Brands")
Out[11]: Text(0.5,1,'Number of Reviews for Top 20 Brands')
Number of Reviews for Top 20 Brands
In [12]:
# Plot number of reviews for top 50 products
products = data["Product Name"].value_counts()
download (1).png
download.png
Feature_Vector_Per....xls
Show all
Search the web and Windows
Links
02:47 AM
09-Jun-20

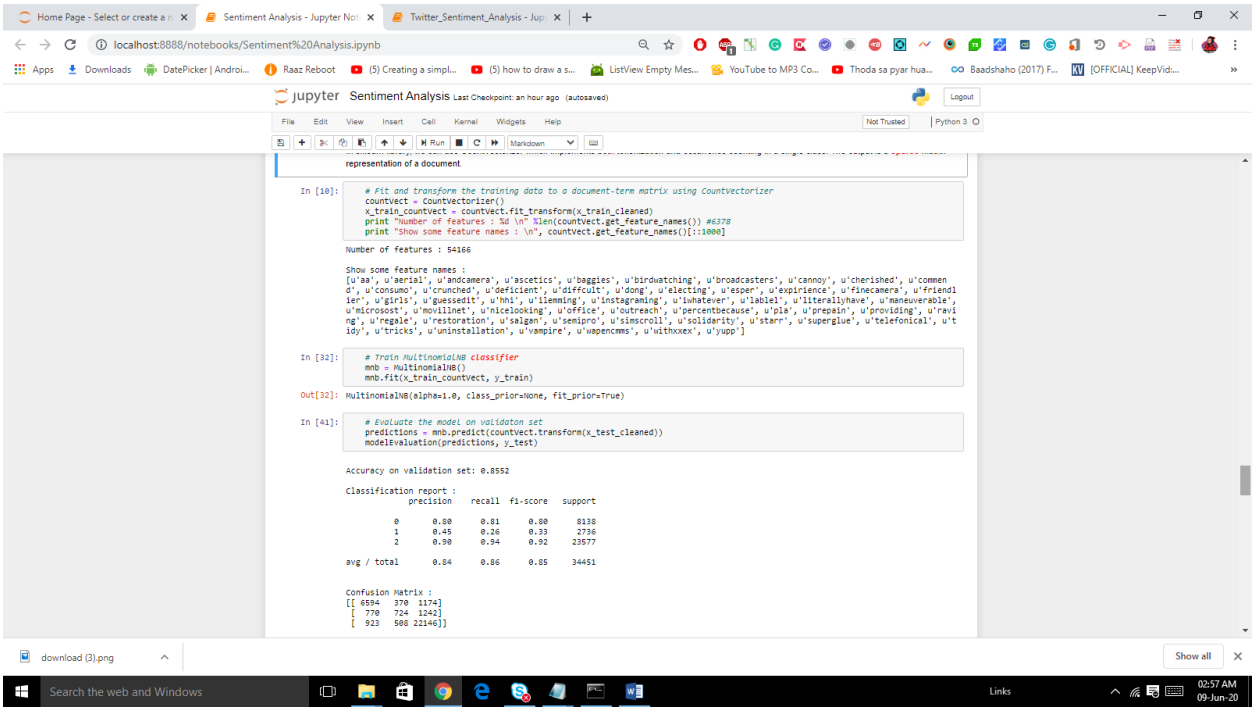

Number of Reviews for Top 20 Brands
60000
50000
40000
30000
20000
10000
0
Samsung
BLU
Apple
LG
BlackBerry
Nokia
Motorola
HTC
CNPGD
OtterBox
Sony
Posh Mobile
Huawei
LG Electronics
samsung
Polaroid
ZTE
Alcatel
Asus
verykool

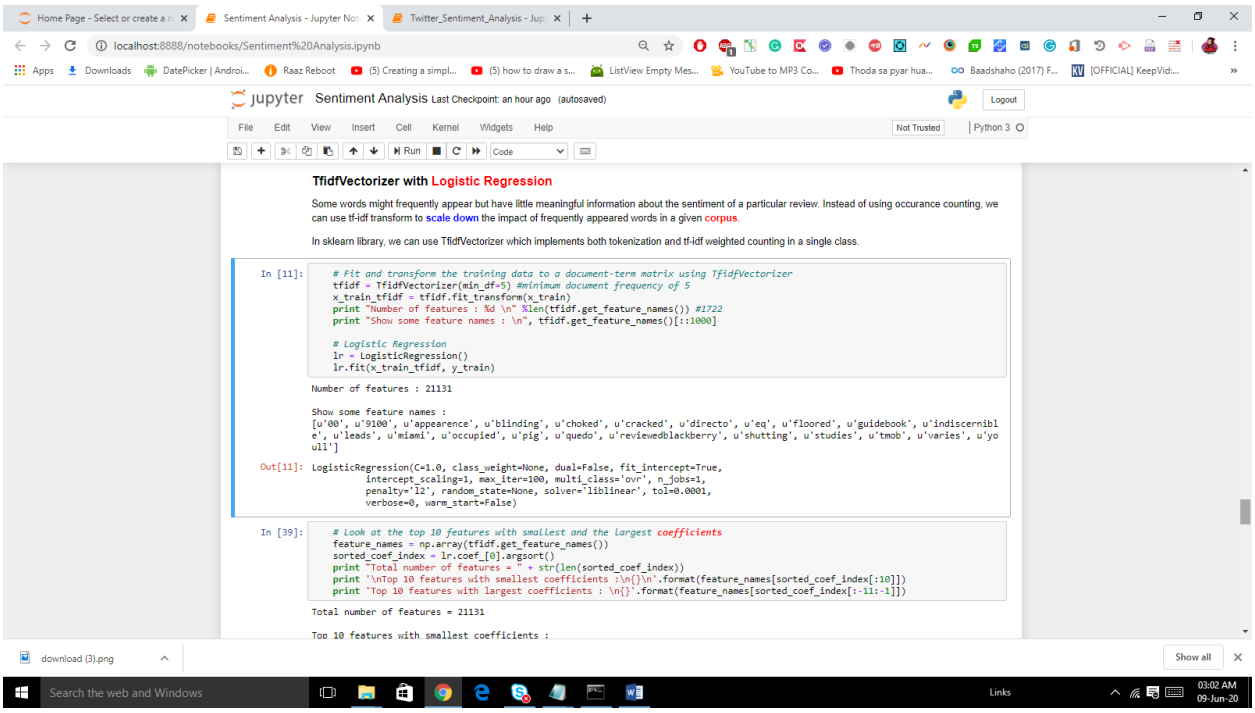

Sentiment Analysis - Jupyter Not
Twitter_Sentiment_Analysis - Jup
localhost:8888/notebooks/Sentiment%20Analysis.ipynb
jupyter Sentiment Analysis Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
In [12]:
# Plot number of reviews for top 50 products
products = data["Product Name"].value_counts()
plt.figure(figsize=(12,8))
products[:50].plot(kind='bar')
plt.title("Number of Reviews for Top 50 Products")
Out[12]:
Text(0.5,1,'Number of Reviews for Top 50 Products')
Number of Reviews for Top 50 Products
1200
1000
800
600
400
200
download (2).png
download (1).png
download.png
Feature_Vector_Per....xls
Show all
Search the web and Windows
Links
02:49 AM
09-Jun-20

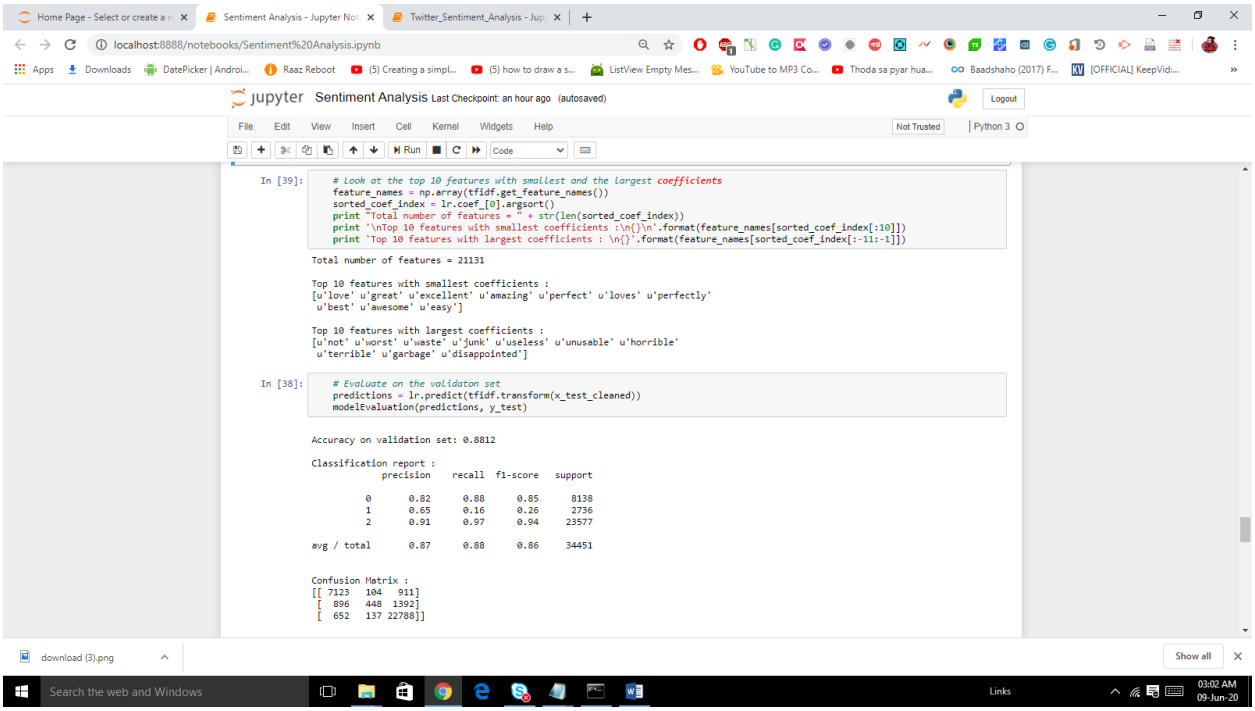

Number of Reviews for Top 50 Products
1200
1000
800
600
400
200
0
BLU Studio 5.0 C HD Unlocked Cellphone, Black
OtterBox Iphone 5/5S/SE Defender Case w/ Drop and Dust Proctection - Realtree AP Pink
Motorola Moto E (1st Generation) - Black - 4 GB - Global GSM Unlocked Phone
Apple iPhone 5s 32GB (Silver) - AT&T
BLU Energy X Plus Smartphone - With 4000 mAh Super Battery- US GSM Unlocked - Grey
Samsung Galaxy S Duos II S7582 DUAL SIM Factory Unlocked International Version - Black
Samsung Galaxy S Duos GT-S7562 GSM Unlocked Touchscreen 5MP Camera Smartphone White
Samsung Galaxy S4 i9505 16GB LTE Unlocked International Version White
Apple iPhone 5s AT&T Cellphone, 16GB, Silver
Apple iPhone 4S 16GB Unlocked GSM - White (Certified Refurbished)
BLU Energy X Plus Smartphone - With 4000 mAh Super Battery- Global GSM Unlocked - Grey
Motorola Moto E (1st Generation) - Black - 4 GB - US GSM Unlocked Phone
Samsung Galaxy S5 SM-G900A GSM Unlocked Cellphone, 16GB, Blue
CNPGD All-in-1 Watch Cell Phone & Smart Watch Sync to Android IOS Smart Phone (Black Plus)
Apple iPhone 5s 32GB (Gold) - AT&T
BLU Dash JR 4.0 K Smartphone - Unlocked - White
Apple iPhone 4 32GB (Black) - AT&T
Huawei Mate 2 - Factory Unlocked (Black)
BlackBerry 8520OEMRED Gemini 8520 Unlocked Phone with 2 MP Camera, Bluetooth, Wi-Fi--International Version (Red)
Samsung Galaxy Exhibit 4G (T-Mobile), t679
BLU Studio 7.0 II -Unlocked Smartphone - US GSM - Blue
Blackberry 9650 Bold Unlocked GSM Smartphone with 3 MP Camera, Bluetooth, 3G, Wi-Fi, and MicroSd Slot (Black)
Samsung Galaxy S7 Edge G9350 32GB HK DUAL SIM Factory Unlocked GSM International Version no warranty (BLUE CORAL)
Blackberry Curve 8520 Unlocked Quad-Band GSM Phone with 2MP Camera, QWERTY Keyboard, Wi-Fi and Bluetooth - Lavender
BLU Dash JR 4.0K Android 4.2, 2MP - Unlocked (Silver)
CNPGD [U.S. Office Extended Warranty] Smartwatch + Unlocked Watch Cell Phone All in 1 Bluetooth Watch for iPhone Android Samsung Galaxy Note,Nexus,htc,Sony White
BLU Studio 7.0 II -Unlocked Smartphone - US GSM - Grey
CNPGD [U.S. Office Extended Warranty] Smartwatch + Unlocked Watch Cell Phone All in 1 Bluetooth Watch for iPhone Android Samsung Galaxy Note,Nexus,htc,Sony Silver
BLU Energy X Plus Smartphone - With 4000 mAh Super Battery- US GSM Unlocked - Silver
CNPGD [U.S. Warranty] All-in-1 Smartwatch and Watch Cell Phone Red
ALCATEL OneTouch Idol 3 Global Unlocked 4G LTE Smartphone, 5.5 HD IPS Display, 16GB (GSM - US Warranty)
Blackberry Curve 8520 Unlocked Quad-Band GSM Phone with 2MP Camera, QWERTY Keyboard, Wi-Fi and Bluetooth - Sky Blue
Blackberry Curve 8520 Gemini SmartPhone Unlocked with 2 MP Camera, Bluetooth, Wi-Fi, E-mail-- International Version with No Warranty (Frost Blue)
Samsung S5830 Galaxy Ace - Unlocked Phone - Black
Samsung Galaxy S5 Mini G800H Unlocked Cellphone, International Version, 16GB, White
Samsung Galaxy S3 SGH-i747 4G LTE GSM Unlocked 16GB No Warranty (White)
CNPGD [U.S. Warranty] All-in-1 Smartwatch and Watch Cell Phone Gold
BLU Studio 5.0 HD Unlocked Cellphone, White
BLU VIVO 5 Smartphone -5.5" 4G LTE GSM Unlocked - 32GB +3GB RAM - Silver
Apple iPhone 4S 16GB (Black) - Verizon
Samsung Galaxy Note I717 16GB 4G LTE GSM Android Phone - Carbon Blue (AT&T version)
BLU Studio Selfie - Smartphone - GSM Unlocked - Gold
BLU Studio 5.0 C HD Smartphone - Unlocked - Orange
OtterBox 77-29864 Defender Series Hybrid Case for Samsung Galaxy Note 10.1 - Retail Packaging - White (2012 Version)
Samsung Galaxy S4 SGH-I337 Unlocked GSM Smartphone w/ 13 MP Camera - Red (No Warranty)
Samsung Galaxy Grand Prime Dual Sim Factory Unlocked Phone - Retail Packaging - Gold(International Version)
Samsung Galaxy S6 G920F 32GB Factory Unlocked GSM 4G LTE Octa-Core Smartphone - Blue Topaz
HTC One M8 - Factory Unlocked 32GB - US Warranty (Glacial Silver)
Apple iPhone 5c 8GB Unlocked GSM Smartphone - Green
Apple iPhone 6 Plus 64GB Gold (Factory Unlocked)

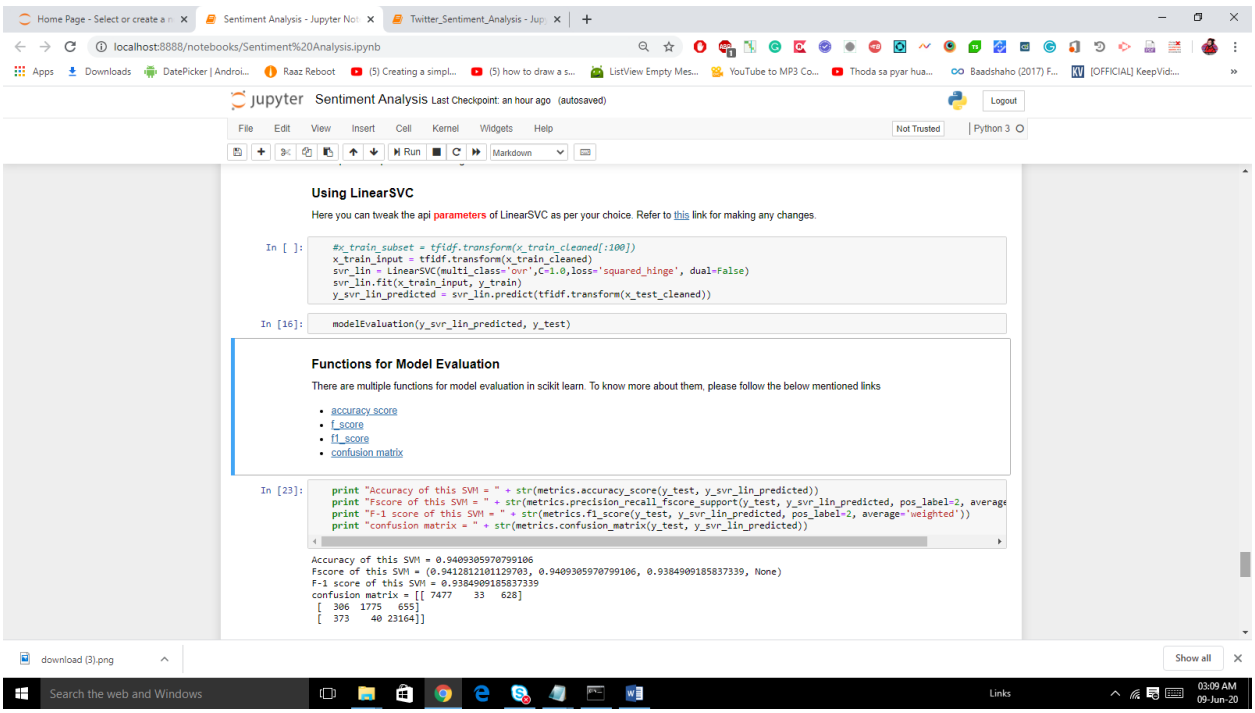

Sentiment Analysis - Jupyter Not
Twitter_Sentiment_Analysis - Jup
localhost:8888/notebooks/Sentiment%20Analysis.ipynb
jupyter Sentiment Analysis Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
In [13]:
# Plot distribution of review length
review_length = data["Reviews"].dropna().map(lambda x: len(x))
plt.figure(figsize=(12,8))
review_length.loc[review_length < 1500].hist()
plt.title("Distribution of Review Length")
plt.xlabel('Review length (Number of character)')
plt.ylabel('Count')
Out[13]: Text(0,0.5,'Count')
Distribution of Review Length
Count
Review length (Number of character)
In [8]:
#split data into training and testing set
x_train, x_test, y_train, y_test = train_test_split(data['Reviews'], data['Label'], test_size=0.1, random_state=0)
Search the web and Windows
Links
02:50 AM
09-Jun-20

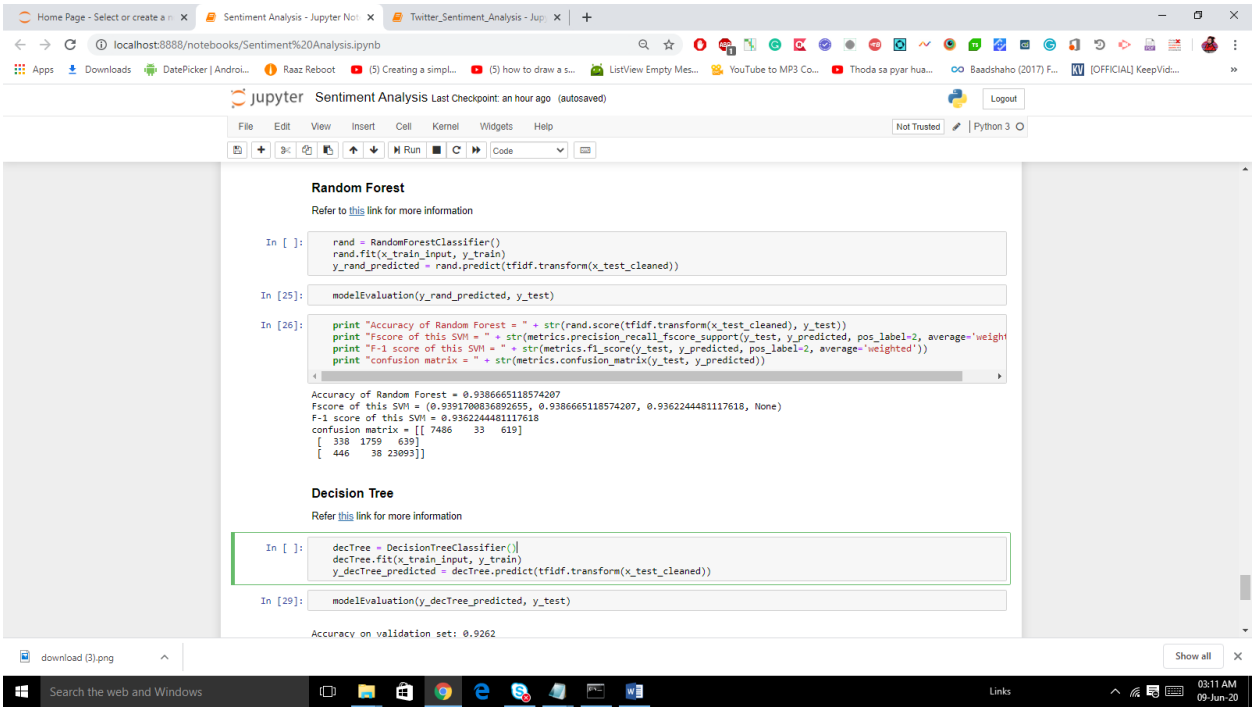

Distribution of Review Length
Count
Review length (Number of character)
0
50000
100000
150000
200000
0
200
400
600
800
1000
1200
1400

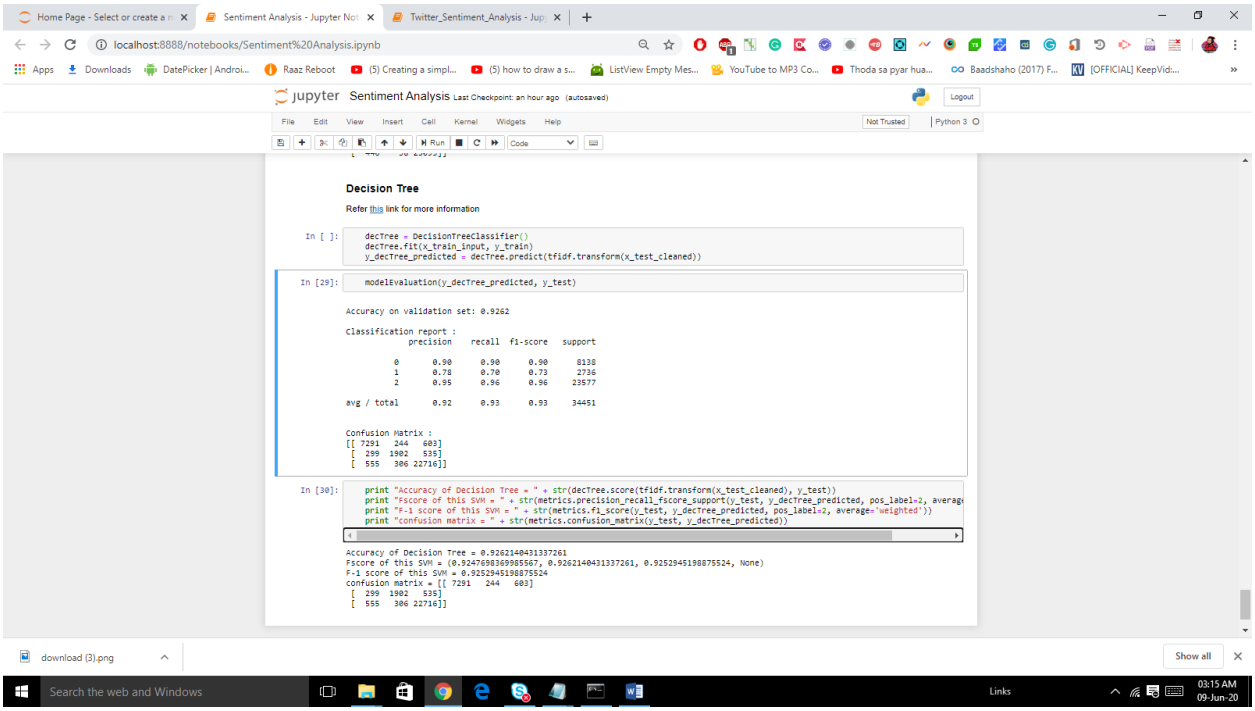

Sentiment Analysis - Jupyter Not
Twitter_Sentiment_Analysis - Jup
localhost:8888/notebooks/Sentiment%20Analysis.ipynb
jupyter Sentiment Analysis Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
representation of a document.
In [10]:
# Fit and transform the training data to a document-term matrix using CountVectorizer
countVect = CountVectorizer()
x_train_countVect = countVect.fit_transform(x_train_cleaned)
print "Number of features : %d \n" %len(countVect.get_feature_names()) #63378
print "Show some feature names : \n", countVect.get_feature_names()[::1000]
Number of features : 54166
Show some feature names :
[u'aa', u'aerial', u'andcamera', u'ascetics', u'baggies', u'birdwatching', u'broadcasters', u'cannoy', u'cherished', u'commend', u'consumo', u'crunched', u'deficient', u'difficult', u'dong', u'electing', u'esper', u'expirience', u'finecamera', u'friendl
ier', u'girls', u'guessedit', u'hhl', u'ilemming', u'instagraming', u'iwhatever', u'label', u'literallyhave', u'maneuverable',
u'microsost', u'movillnet', u'nicelooking', u'office', u'outreach', u'percentbecause', u'pla', u'prepain', u'providing', u'ravi
ng', u'regale', u'restoration', u'salgan', u'semipro', u'simscroll', u'solidarity', u'starr', u'superglue', u'telefonica', u't
idy', u'tricks', u'uninstallation', u'vampire', u'wapencmms', u'withxxex', u'yupp']
In [32]:
# Train MultinomialNB classifier
mnb = MultinomialNB()
mnb.fit(x_train_countVect, y_train)
Out[32]: MultinomialNB(alpha=1.0, class_prior=None, fit_prior=True)
In [41]:
# Evaluate the model on validaton set
predictions = mnb.predict(countVect.transform(x_test_cleaned))
modelEvaluation(predictions, y_test)
Accuracy on validation set: 0.8552
Classification report :
precision recall f1-score support
0 0.80 0.81 0.80 8138
1 0.45 0.26 0.33 2736
2 0.90 0.94 0.92 23577
avg / total 0.84 0.86 0.85 34451
Confusion Matrix :
[[6594 370 1174]
[770 724 1242]
[923 508 22146]]
download (3).png
Show all
Search the web and Windows
Links
02:57 AM
09-Jun-20


Sentiment Analysis - Jupyter Not
Twitter_Sentiment_Analysis - Jup
localhost:8888/notebooks/Sentiment%20Analysis.ipynb
jupyter Sentiment Analysis Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
TfidfVectorizer with Logistic Regression
Some words might frequently appear but have little meaningful information about the sentiment of a particular review. Instead of using occurance counting, we can use tf-idf transform to scale down the impact of frequently appeared words in a given corpus.
In sklearn library, we can use TfidfVectorizer which implements both tokenization and tf-idf weighted counting in a single class.
In [11]:
# Fit and transform the training data to a document-term matrix using TfidfVectorizer
tfidf = TfidfVectorizer(min_df=5) #minimum document frequency of 5
x_train_tfidf = tfidf.fit_transform(x_train)
print "Number of features : %d \n" %len(tfidf.get_feature_names()) #1722
print "Show some feature names : \n", tfidf.get_feature_names()[::1000]
# Logistic Regression
lr = LogisticRegression()
lr.fit(x_train_tfidf, y_train)
Number of features : 21131
Show some feature names :
[u'00', u'9100', u'appearence', u'blinding', u'choked', u'cracked', u'directo', u'eq', u'floored', u'guidebook', u'indiscernibl
e', u'leads', u'miami', u'occupied', u'pig', u'quedo', u'reviewedblackberry', u'shutting', u'studies', u'tmob', u'varies', u'yo
ull']
Out[11]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
verbose=0, warm_start=False)
In [39]:
# Look at the top 10 features with smallest and the largest coefficients
feature_names = np.array(tfidf.get_feature_names())
sorted_coef_index = lr.coef_[0].argsort()
print 'Total number of features = ' + str(len(sorted_coef_index))
print '\nTop 10 features with smallest coefficients :\n{}\n'.format(feature_names[sorted_coef_index[:10]])
print 'Top 10 features with largest coefficients : \n{}'.format(feature_names[sorted_coef_index[:-11:-1]])
Total number of features = 21131
Top 10 features with smallest coefficients :
download (3).png
Show all
Search the web and Windows
Links
03:02 AM
09-Jun-20


Sentiment Analysis - Jupyter Not
Twitter_Sentiment_Analysis - Jup
localhost:8888/notebooks/Sentiment%20Analysis.ipynb
jupyter Sentiment Analysis Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
In [39]:
# Look at the top 10 features with smallest and the largest coefficients
feature_names = np.array(tfidf.get_feature_names())
sorted_coef_index = lr.coef_[0].argsort()
print "Total number of features = " + str(len(sorted_coef_index))
print '\nTop 10 features with smallest coefficients :\n{}\n'.format(feature_names[sorted_coef_index[:10]])
print 'Top 10 features with largest coefficients : \n{}'.format(feature_names[sorted_coef_index[:-11:-1]])
Total number of features = 21131
Top 10 features with smallest coefficients :
[u'love' u'great' u'excellent' u'amazing' u'perfect' u'loves' u'perfectly'
u'best' u'awesome' u'easy']
Top 10 features with largest coefficients :
[u'not' u'worst' u'waste' u'junk' u'useless' u'unusable' u'horrible'
u'terrible' u'garbage' u'disappointed']
In [38]:
# Evaluate on the validaton set
predictions = lr.predict(tfidf.transform(x_test_cleaned))
modelEvaluation(predictions, y_test)
Accuracy on validation set: 0.8812
Classification report :
precision recall f1-score support
0 0.82 0.88 0.85 8138
1 0.65 0.16 0.26 2736
2 0.91 0.97 0.94 23577
avg / total 0.87 0.88 0.86 34451
Confusion Matrix :
[[7123 104 911]
[896 448 1392]
[652 137 22788]]
download (3).png
Show all
Search the web and Windows
Links
03:02 AM
09-Jun-20

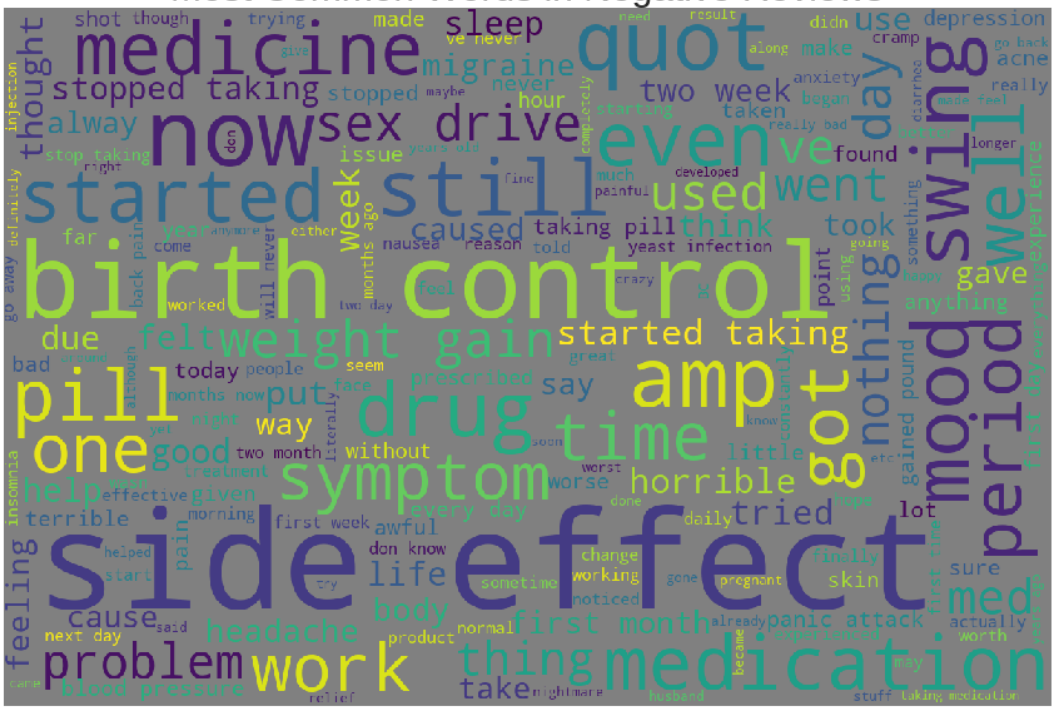

Sentiment Analysis - Jupyter Not
Twitter_Sentiment_Analysis - Jup
localhost:8888/notebooks/Sentiment%20Analysis.ipynb
jupyter Sentiment Analysis Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Markdown
Using LinearSVC
Here you can tweak the api parameters of LinearSVC as per your choice. Refer to this link for making any changes.
In [ ]:
#x_train_subset = tfidf.transform(x_train_cleaned[:100])
x_train_input = tfidf.transform(x_train_cleaned)
svr_lin = LinearSVC(multi_class='ovr',C=1.0,loss='squared_hinge', dual=False)
svr_lin.fit(x_train_input, y_train)
y_svr_lin_predicted = svr_lin.predict(tfidf.transform(x_test_cleaned))
In [16]:
modelEvaluation(y_svr_lin_predicted, y_test)
Functions for Model Evaluation
There are multiple functions for model evaluation in scikit learn. To know more about them, please follow the below mentioned links
accuracy score
f_score
f1_score
confusion matrix
In [23]:
print "Accuracy of this SVM = " + str(metrics.accuracy_score(y_test, y_svr_lin_predicted))
print "Fscore of this SVM = " + str(metrics.precision_recall_fscore_support(y_test, y_svr_lin_predicted, pos_label=2, average
print "F-1 score of this SVM = " + str(metrics.f1_score(y_test, y_svr_lin_predicted, pos_label=2, average='weighted'))
print "confusion matrix = " + str(metrics.confusion_matrix(y_test, y_svr_lin_predicted))
Accuracy of this SVM = 0.9409305970799106
Fscore of this SVM = (0.9412812101129703, 0.9409305970799106, 0.9384909185837339, None)
F-1 score of this SVM = 0.9384909185837339
confusion matrix = [[7477 33 628]
[306 1775 655]
[373 40 23164]]
download (3).png
Show all
Search the web and Windows
Links
03:09 AM
09-Jun-20

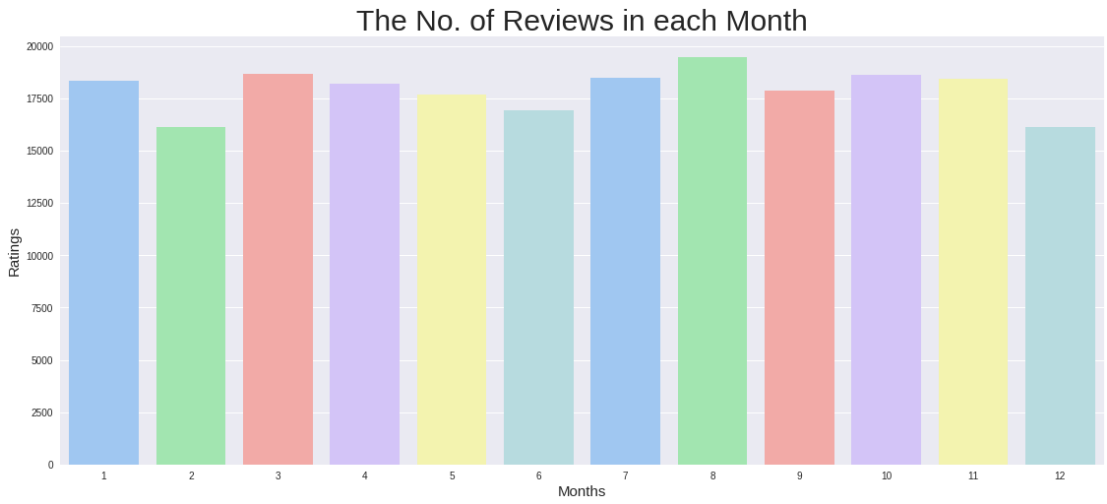

Sentiment Analysis - Jupyter Not
Twitter_Sentiment_Analysis - Jup
localhost:8888/notebooks/Sentiment%20Analysis.ipynb
jupyter Sentiment Analysis Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
Random Forest
Refer to this link for more information
In [ ]:
rand = RandomForestClassifier()
rand.fit(x_train_input, y_train)
y_rand_predicted = rand.predict(tfidf.transform(x_test_cleaned))
In [25]:
modelEvaluation(y_rand_predicted, y_test)
In [26]:
print "Accuracy of Random Forest = " + str(rand.score(tfidf.transform(x_test_cleaned), y_test))
print "Fscore of this SVM = " + str(metrics.precision_recall_fscore_support(y_test, y_predicted, pos_label=2, average='weight
print "F-1 score of this SVM = " + str(metrics.f1_score(y_test, y_predicted, pos_label=2, average='weighted'))
print "confusion matrix = " + str(metrics.confusion_matrix(y_test, y_predicted))
Accuracy of Random Forest = 0.9386665118574207
Fscore of this SVM = (0.9391700836892655, 0.9386665118574207, 0.9362244481117618, None)
F-1 score of this SVM = 0.9362244481117618
confusion matrix = [[7486 33 619]
[338 1759 639]
[446 38 23093]]
Decision Tree
Refer this link for more information
In [ ]:
decTree = DecisionTreeClassifier()
decTree.fit(x_train_input, y_train)
y_decTree_predicted = decTree.predict(tfidf.transform(x_test_cleaned))
In [29]:
modelEvaluation(y_decTree_predicted, y_test)
Accuracy on validation set: 0.9262
download (3).png
Show all
Search the web and Windows
Links
03:11 AM
09-Jun-20

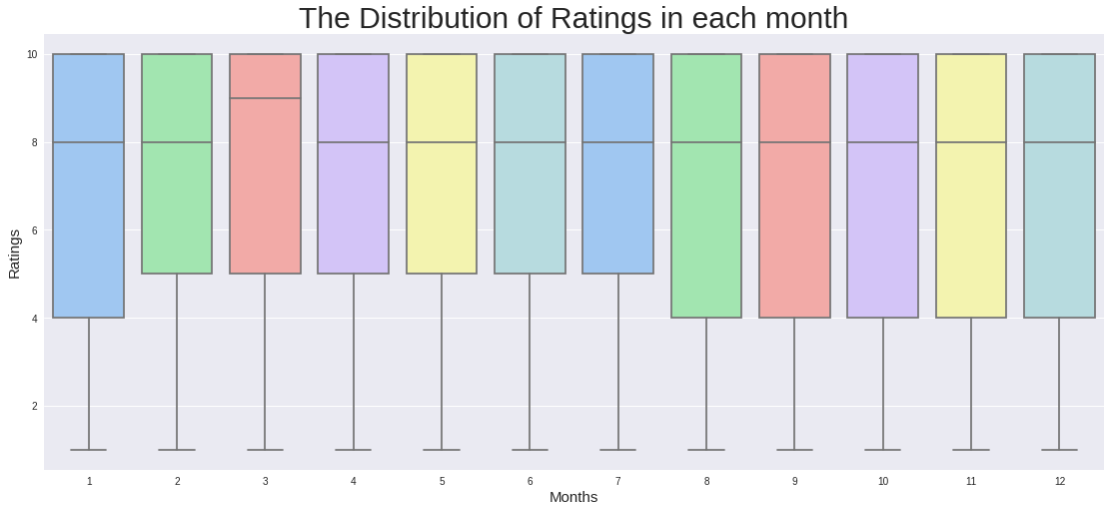

Sentiment Analysis - Jupyter Not
Twitter_Sentiment_Analysis - Jup
localhost:8888/notebooks/Sentiment%20Analysis.ipynb
jupyter Sentiment Analysis Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3
Run
Code
Decision Tree
Refer this link for more information
In [ ]:
decTree = DecisionTreeClassifier()
decTree.fit(x_train_input, y_train)
y_decTree_predicted = decTree.predict(tfidf.transform(x_test_cleaned))
In [29]:
modelEvaluation(y_decTree_predicted, y_test)
Accuracy on validation set: 0.9262
Classification report :
precision recall f1-score support
0 0.90 0.90 0.90 8138
1 0.78 0.70 0.73 2736
2 0.95 0.96 0.96 23577
avg / total 0.92 0.93 0.93 34451
Confusion Matrix :
[[7291 244 603]
[299 1902 535]
[555 306 22716]]
In [30]:
print "Accuracy of Decision Tree = " + str(decTree.score(tfidf.transform(x_test_cleaned), y_test))
print "Fscore of this SVM = " + str(metrics.precision_recall_fscore_support(y_test, y_decTree_predicted, pos_label=2, average
print "F-1 score of this SVM = " + str(metrics.f1_score(y_test, y_decTree_predicted, pos_label=2, average='weighted'))
print "confusion matrix = " + str(metrics.confusion_matrix(y_test, y_decTree_predicted))
Accuracy of Decision Tree = 0.9262140431337261
Fscore of this SVM = (0.9247698369985567, 0.9262140431337261, 0.9252945198875524, None)
F-1 score of this SVM = 0.9252945198875524
confusion matrix = [[7291 244 603]
[299 1902 535]
[555 306 22716]]
download (3).png
Show all
Search the web and Windows
Links
03:15 AM
09-Jun-20

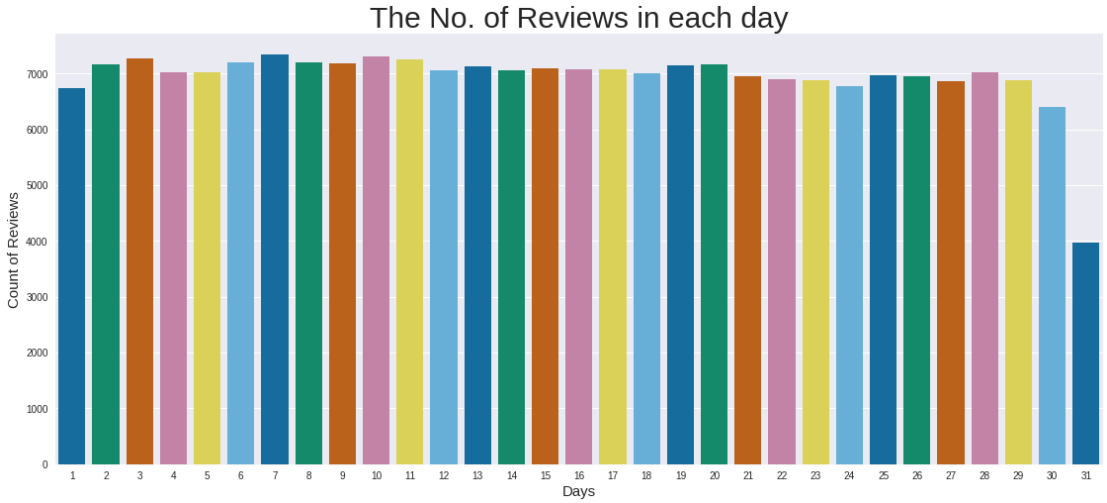
Most Common Words n Positive Reviews

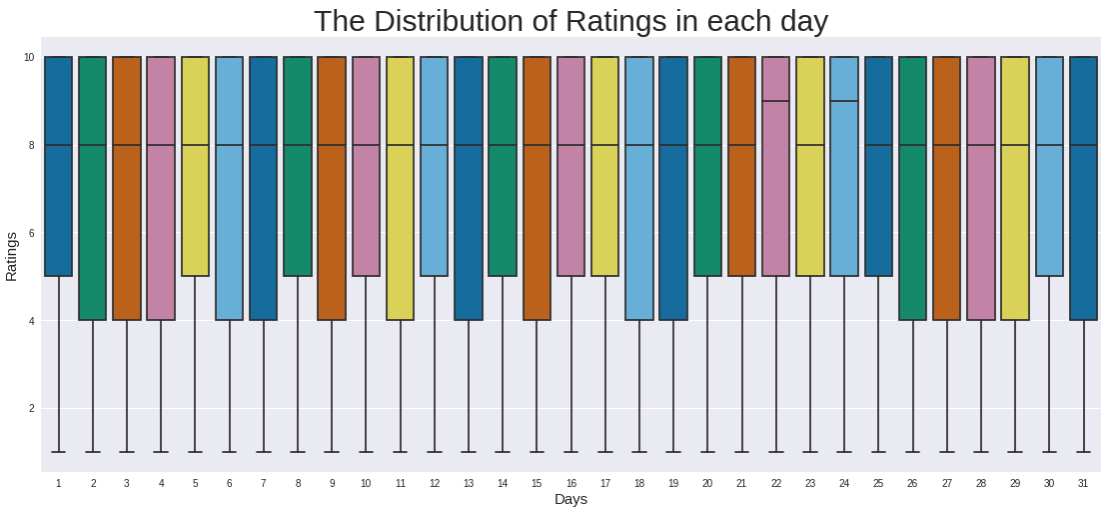
Most Common Words in Negative Reviews

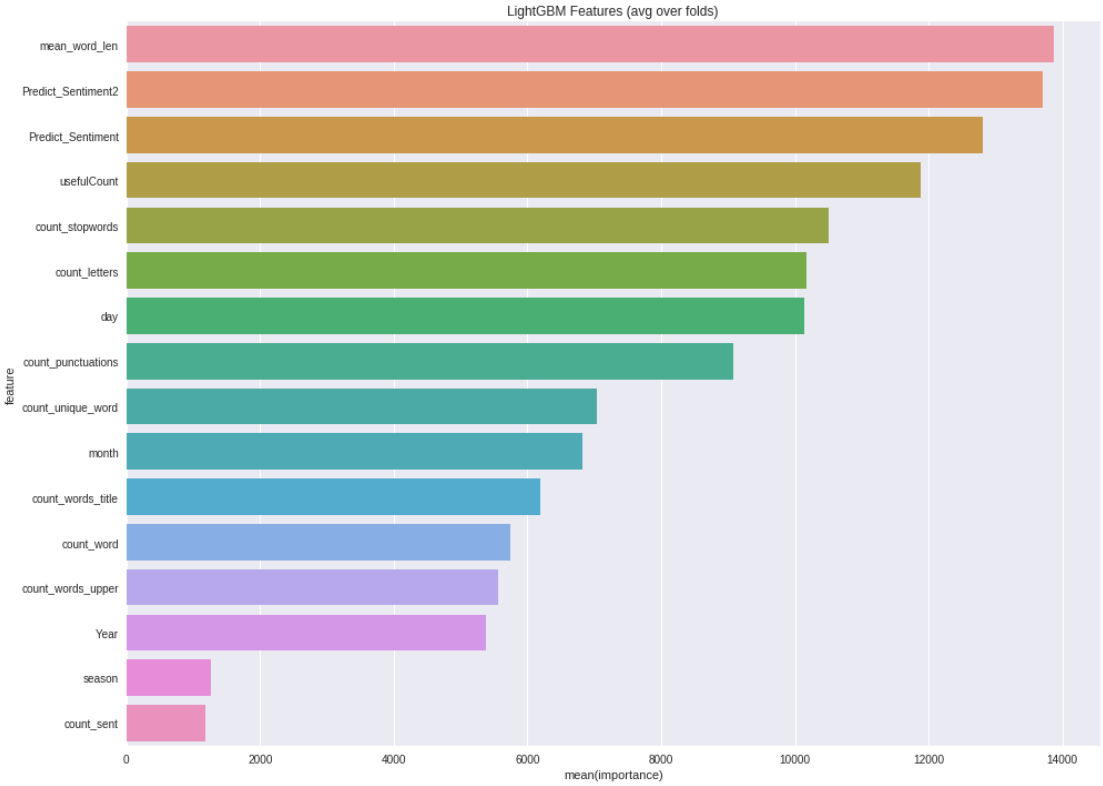


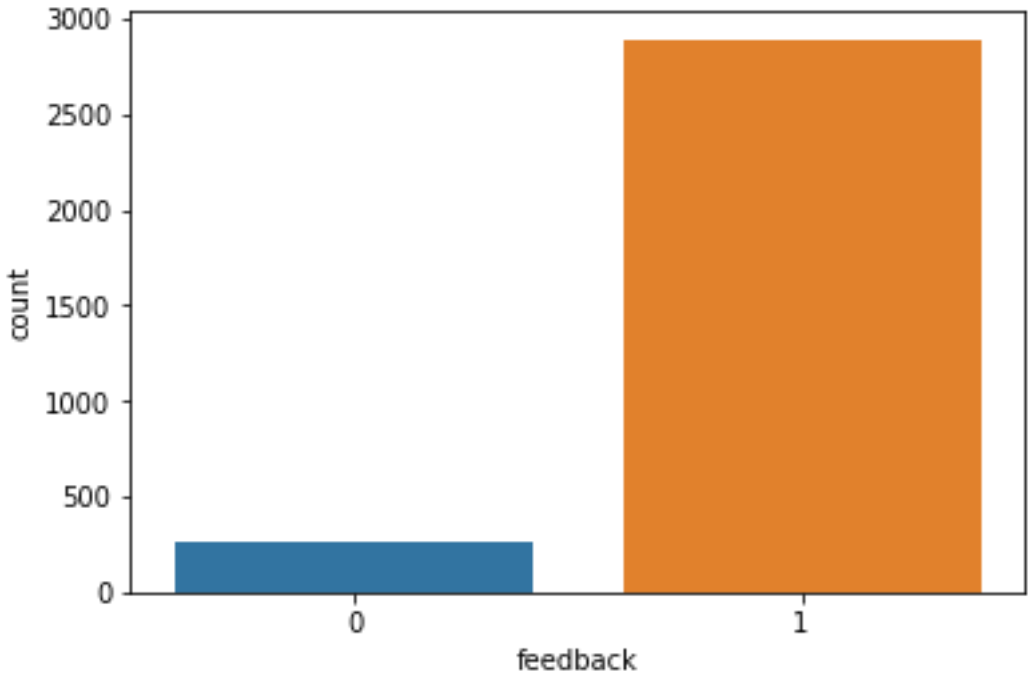

The No. of Reviews in each Month
Ratings
Months
0
2500
5000
7500
10000
12500
15000
17500
20000
1
2
3
4
5
6
7
8
9
10
11
12

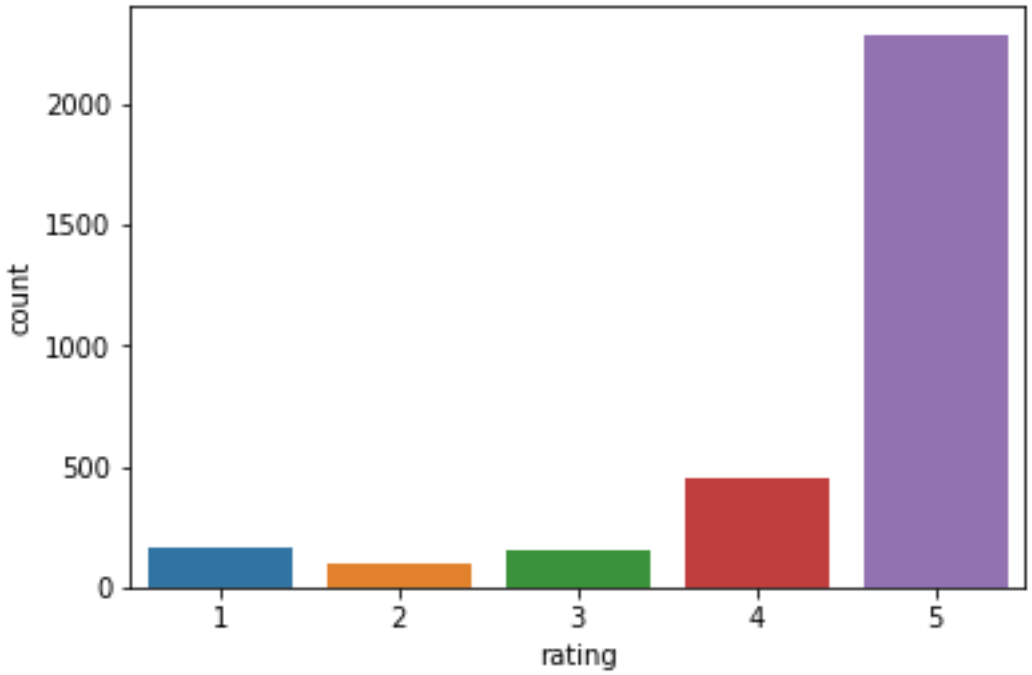

The Distribution of Ratings in each month
Ratings
Months
2
4
6
8
10
1
2
3
4
5
6
7
8
9
10
11
12

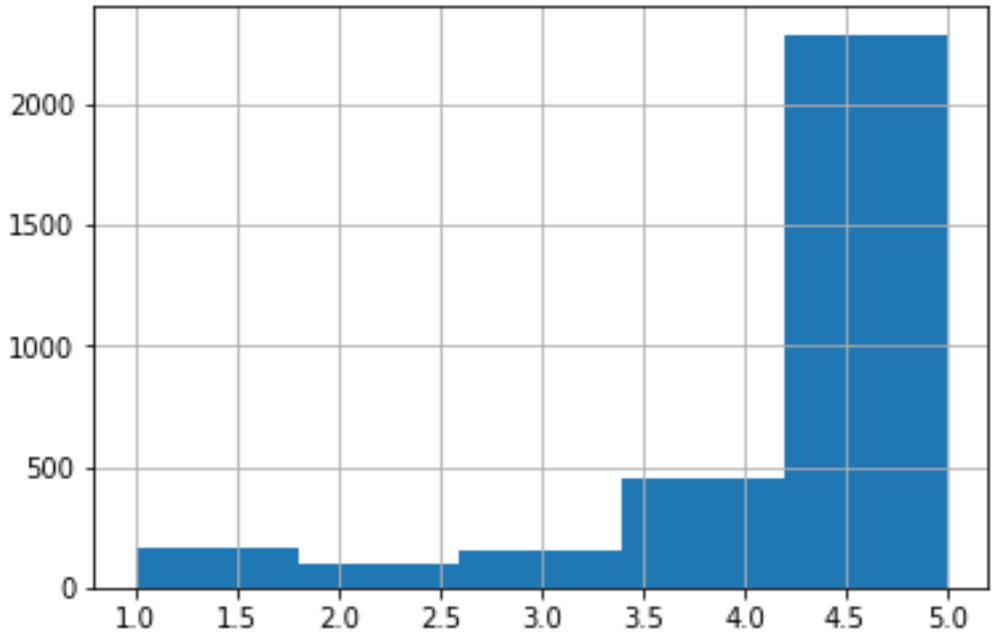

The No. of Reviews in each day
Count of Reviews
Days
0
1000
2000
3000
4000
5000
6000
7000
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

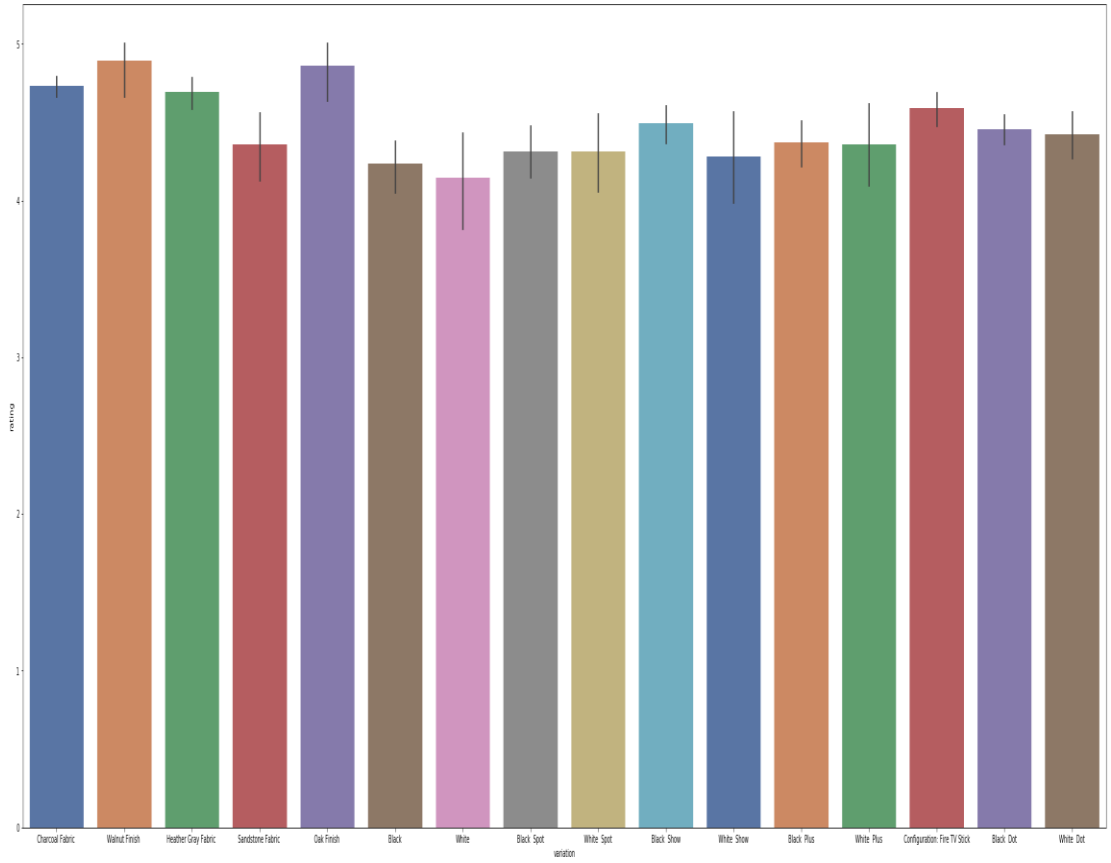

The Distribution of Ratings in each day
Ratings
Days

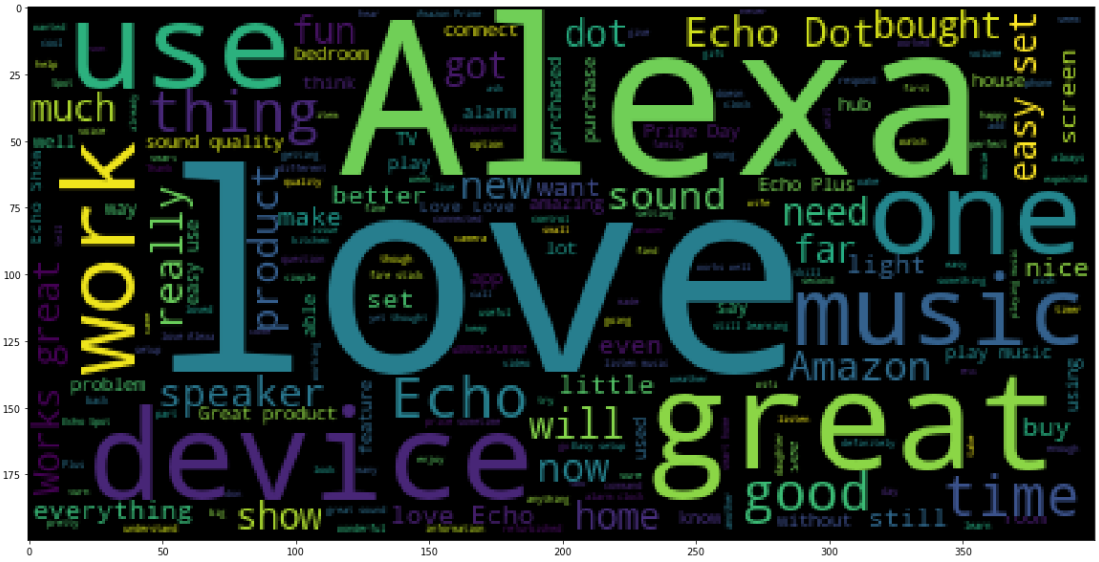

LightGBM Features (avg over folds)
mean_word_len
Predict_Sentiment2
Predict_Sentiment
usefulCount
count_stopwords
count_letters
day
count_punctuations
count_unique_word
month
count_words_title
count_word
count_words_upper
Year
season
count_sent
feature
mean(importance)


3000
2500
2000
1500
1000
500
0
count
0
1
feedback

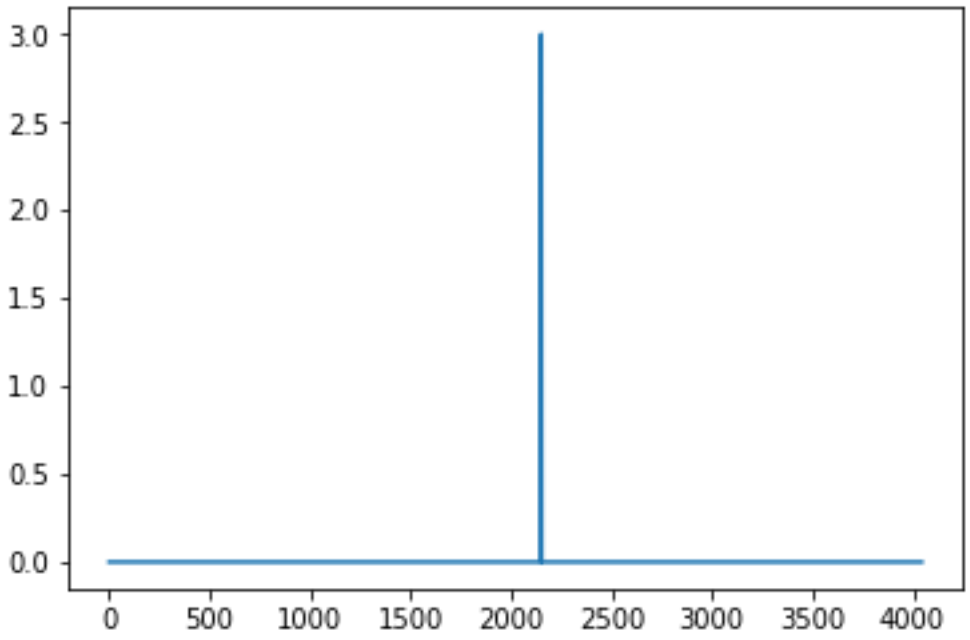

2000
1500
1000
500
0
count
1
2
3
4
5
rating

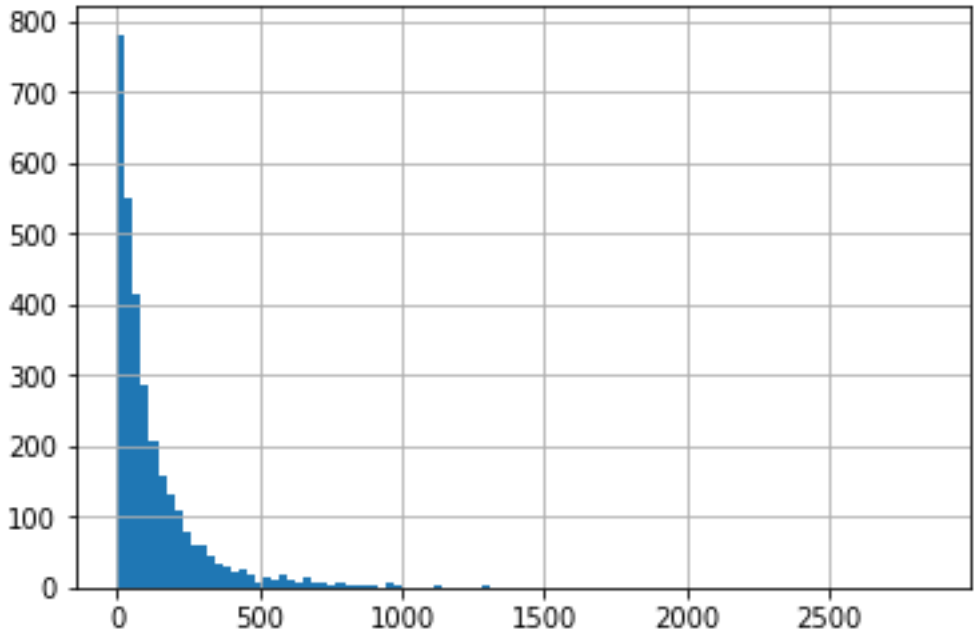

2000
1500
1000
500
0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


rating
variation
Charcoal Fabric
Walnut Finish
Heather Gray Fabric
Sandstone Fabric
Oak Finish
Black
White
Black Spot
White Spot
Black Show
White Show
Black Plus
White Plus
Configuration: Fire TV Stick
Black Dot
White Dot



Data Cleaning and Feature Engineering






```
Classification Report              precision    recall  f1-score   support

           0       0.95      0.29      0.45        68
           1       0.94      1.00      0.97       720

    accuracy                           0.94       788
   macro avg       0.94      0.65      0.71       788
weighted avg       0.94      0.94      0.92       788

Confusion Matrix [[ 20  48]
 [  1 719]]
Accuracy Score 0.9378172588832487
Precision Score 0.9374185136897001
Recall Score 0.9986111111111111
F1 Score 0.9670477471418965
```

```
Classification Report              precision    recall  f1-score   support

           0       1.00      0.03      0.06        68
           1       0.92      1.00      0.96       720

    accuracy                           0.92       788
   macro avg       0.96      0.51      0.51       788
weighted avg       0.92      0.92      0.88       788

Confusion Matrix [[  2  66]
 [  0 720]]
Accuracy Score 0.916243654822335
Precision Score 0.916030534351145
Recall Score 1.0
F1 Score 0.9561752988047809
```

Original Naive Bayes

	precision	recall	f1-score	support
positive	0.59	0.78	0.67	2979
negative	0.68	0.46	0.55	3022
accuracy			0.62	6001
macro avg	0.64	0.62	0.61	6001
weighted avg	0.64	0.62	0.61	6001

Multinomail naive bayes

	precision	recall	f1-score	support
positive	0.59	0.79	0.67	2979
negative	0.69	0.46	0.55	3022
accuracy			0.62	6001
macro avg	0.64	0.62	0.61	6001
weighted avg	0.64	0.62	0.61	6001

Logistic Regression

Logistic regression

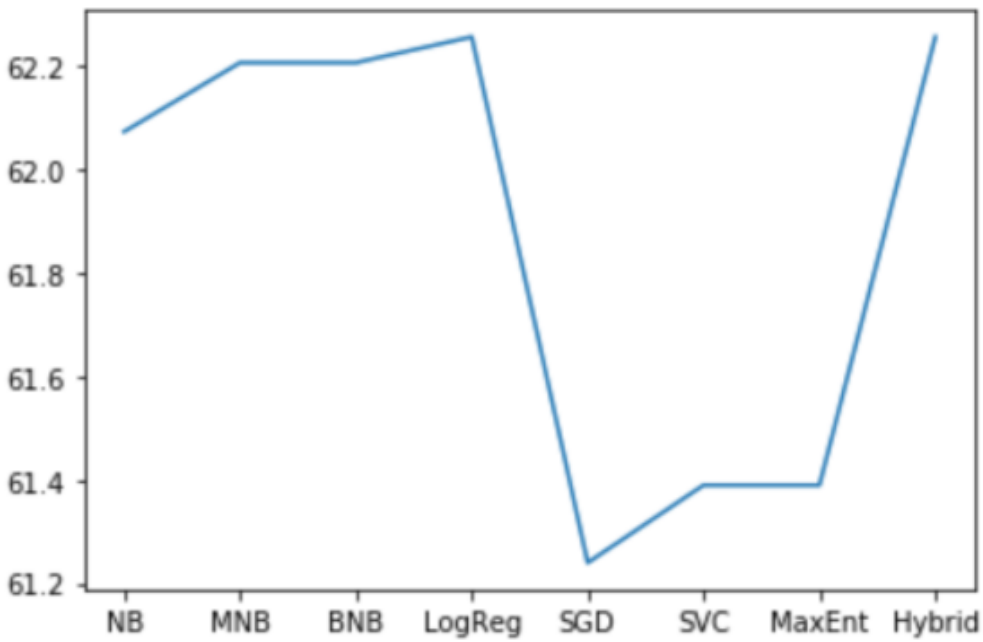
	precision	recall	f1-score	support
positive	0.59	0.78	0.67	2979
negative	0.68	0.47	0.55	3022
accuracy			0.62	6001
macro avg	0.64	0.62	0.61	6001
weighted avg	0.64	0.62	0.61	6001

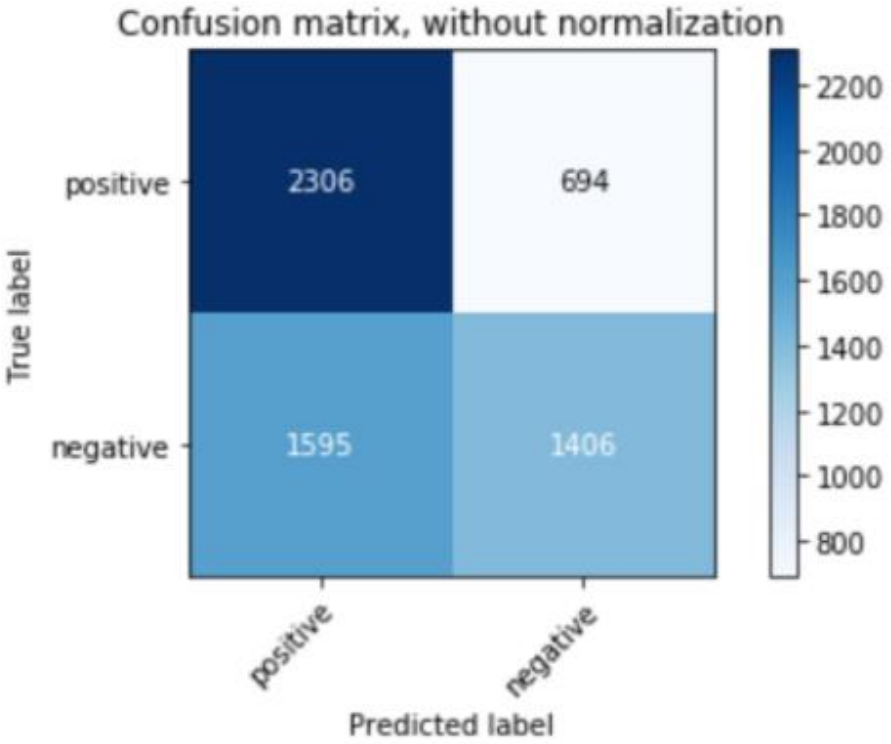
Stochastic Gradient Descent

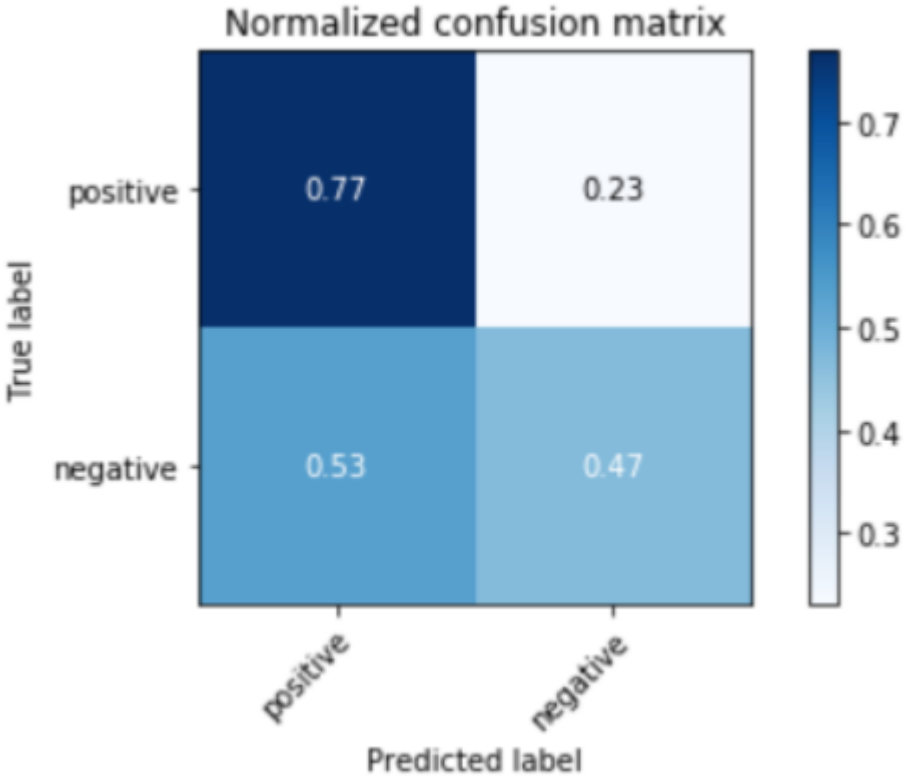
	precision	recall	f1-score	support
positive	0.69	0.40	0.51	2979
negative	0.58	0.82	0.68	3022
accuracy			0.61	6001
macro avg	0.64	0.61	0.59	6001
weighted avg	0.63	0.61	0.59	6001

Support vector classifier

	precision	recall	f1-score	support
positive	0.58	0.79	0.67	2979
negative	0.68	0.44	0.53	3022
accuracy			0.61	6001
macro avg	0.63	0.62	0.60	6001
weighted avg	0.63	0.61	0.60	6001




Confusion matrix, without normalization
positive
negative
True label
2306
694
1595
1406
2200
2000
1800
1600
1400
1200
1000
800
positive
negative
Predicted label


Normalized confusion matrix
positive
negative
True label
0.77
0.23
0.53
0.47
0.7
0.6
0.5
0.4
0.3
positive
negative
Predicted label