**Detection of Cross-Site Scripting (XSS) Using Machine Learning Methods**

**Abstract**

* This article discusses the problem of detecting cross-site scripting (XSS) using machine learning methods. XSS is an attack in which malicious code is embedded on a page to interact with an attacker's web server.
* The XSS attack ranks third in the ranking of key web application risks according to Open Source Foundation for Application Security (OWASP). This attack has not been studied for a long time.
* It was considered harmless. However, this is fallacious: the page or HTTP Cookie may contain very vulnerable data, such as payment document numbers or the administrator session token.
* Machine learning is a tool that can be used to detect XSS attacks. This article describes an experiment. As a result the model for detecting XSS attacks was created.
* In this project we are using a machine learning algorithm, which is Multinomial Naive Bayes classifier.

**Introduction**

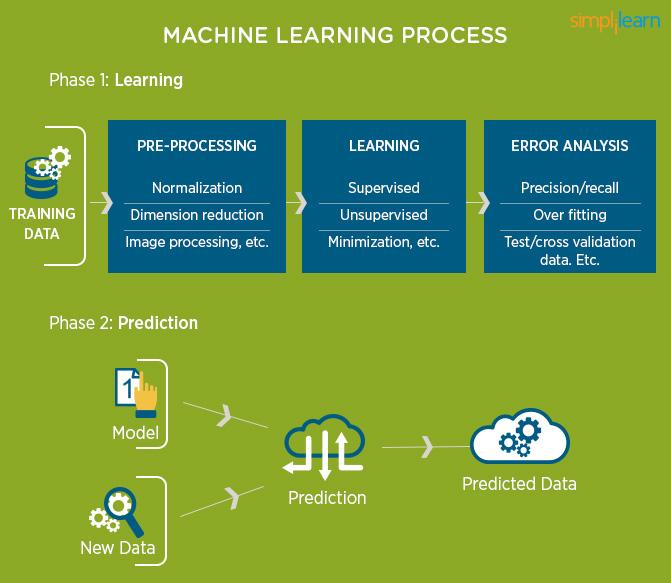
* Web applications are widely used and include sensitive and personal data. This makes them a target for malware that uses vulnerabilities to get unauthorized data stored on the computer.
* Such attacks include cross-site scripting (XSS). Cross-Site scripting (XSS) is a type of attack on web systems that involves inserting malicious code into a page issued by a web system (which will be executed on the user's computer when they open this page) and interacting this code with the attacker's web server.
* This is a type of "code Injection" attack. The specific feature of such attacks is that malicious code can use the user's authorization in the web system to get extended access to it or to get the user's authorization data.
* Malicious code can be inserted into a page either through a vulnerability in the web server or through a vulnerability on the user's computer. The term is abbreviated "XSS" to avoid confusion with cascading style sheets that use the abbreviation "CSS".
* XSS can affect the victim by stealing cookies, changing the web page, capturing clipboard content, scanning ports, or dynamically loading. Therefore, the security of web applications is a very important task for developers.
* The lack of client input verification is the most common security flaw in web applications. These shortcomings are repeatedly detected and exploited both on the client side and on the server side. XSS attacks remain in the top ten vulnerabilities listed in the Open Web Application Security Project (OWASP). XSS attacks have not been studied for a long time.
* Such attacks were considered not dangerous. However, this opinion is considered erroneous now. On a site where there is no protection against cross-site request forgery (CSRF), an attacker can perform any actions available to the user.
* But even with this protection, the page or HTTP Cookie may contain very vulnerable data (for example, the administrator session ID or payment document numbers). Cross-site scripting can be used to conduct a DoS attack

## WHY MACHINE LEARNING

To better understand the uses of [machine learning](http://www.cs.princeton.edu/courses/archive/spr08/cos511/scribe_notes/0204.pdf), consider some of the instances where machine learning is applied: the self-driving Google car, cyber fraud detection, online recommendation engines—like friend suggestions on Facebook, Netflix showcasing the movies and shows you might like, and “more items to consider” and “get yourself a little something” on Amazon—are all examples of applied machine learning.

All these examples echo the vital role machine learning has begun to take in today’s data-rich world. Machines can aid in filtering useful pieces of information that help in major advancements, and we are already seeing how this technology is being implemented in a wide variety of industries.

The process flow depicted here represents how machine learning works



With the constant evolution of the field, there has been a subsequent rise in the uses, demands, and importance of machine learning. Big data has become quite a buzzword in the last few years; that’s in part due to increased sophistication of machine learning, which helps analyze those big chunks of big data. Machine learning has also changed the way data extraction, and interpretation is done by involving automatic sets of generic methods that have replaced traditional statistical techniques.

## Uses Of Machine Learning

Earlier in this article, we mentioned some applications of machine learning. To understand the concept of machine learning better, let’s consider some more examples: web search results, real-time ads on web pages and mobile devices, email spam filtering, network intrusion detection, and pattern and image recognition. All these are by-products of applying machine learning to analyze huge volumes of data.

Traditionally, data analysis was always being characterized by trial and error, an approach that becomes impossible when data sets are large and heterogeneous. Machine learning comes as the solution to all this chaos by proposing clever alternatives to analyzing huge volumes of data. By developing fast and efficient algorithms and data-driven models for real-time processing of data, machine learning is able to produce accurate results and analysis.

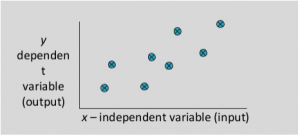
Machine learning tasks are classified into several broad categories. In [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning), the algorithm builds a mathematical model of a set of data that contains both the inputs and the desired outputs. For example, if the task were determining whether an image contained a certain object, the [training data](https://en.wikipedia.org/wiki/Training_data) for a supervised learning algorithm would include images with and without that object (the input), and each image would have a label (the output) designating whether it contained the object. In special cases, the input may be only partially available, or restricted to special feedback. [Semi-supervised learning](https://en.wikipedia.org/wiki/Semi-supervised_learning) algorithms develop mathematical models from incomplete training data, where a portion of the sample inputs are missing the desired output.

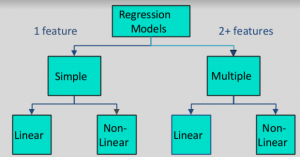
[Classification](https://en.wikipedia.org/wiki/Statistical_classification) algorithms and [regression](https://en.wikipedia.org/wiki/Regression_analysis) algorithms are types of supervised learning. Classification algorithms are used when the outputs are restricted to a [limited set](https://en.wikipedia.org/wiki/Discrete_number) of values. For a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email. For an algorithm that identifies spam emails, the output would be the prediction of either "spam" or "not spam", represented by the [Boolean](https://en.wikipedia.org/wiki/Boolean_data_type) values true and false. [Regression](https://en.wikipedia.org/wiki/Regression_analysis) algorithms are named for their continuous outputs, meaning they may have any value within a range. Examples of a continuous value are the temperature, length, or price of an object.

[Supervised Machine Learning](https://www.geeksforgeeks.org/supervised-unsupervised-learning/)**:**The majority of practical machine learning uses supervised learning. Supervised learning is where you have input variables (x) and an output variable (Y) and you use an algorithm to learn the mapping function from the input to the output**Y = f(X)**. The goal is to approximate the mapping function so well that when you have new input data (x) that you can predict the output variables (Y) for that data.

Techniques of Supervised Machine Learning algorithms include **linear** and  **logistic regression**, **multi-class classification**, **Decision Trees** and **support vector machines**. Supervised learning requires that the data used to train the algorithm is already labeled with correct answers. For example, a classification algorithm will learn to identify animals after being trained on a dataset of images that are properly labeled with the species of the animal and some identifying characteristics.  
Supervised learning problems can be further grouped into **Regression** and  **Classification** problems. Both problems have as goal the construction of a succinct model that can predict the value of the dependent attribute from the attribute variables. The difference between the two tasks is the fact that the dependent attribute is numerical for regression and categorical for classification.

A regression problem is when the output variable is a real or continuous value, such as “salary” or “weight”. Many different models can be used, the simplest is the linear regression. It tries to fit data with the best hyper-plane which goes through the points.

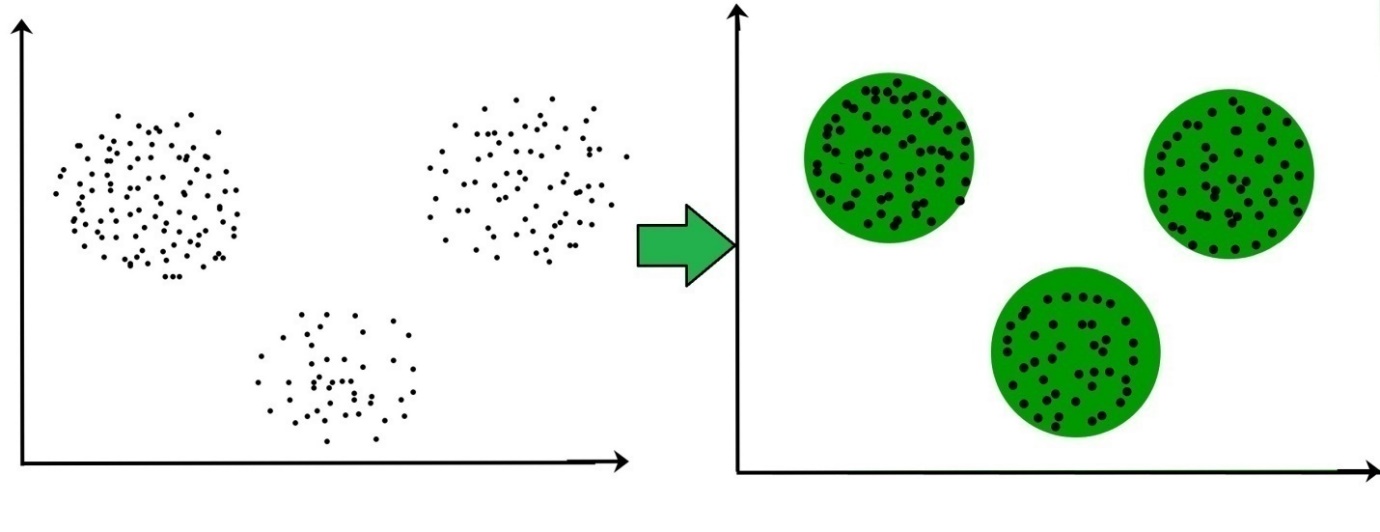


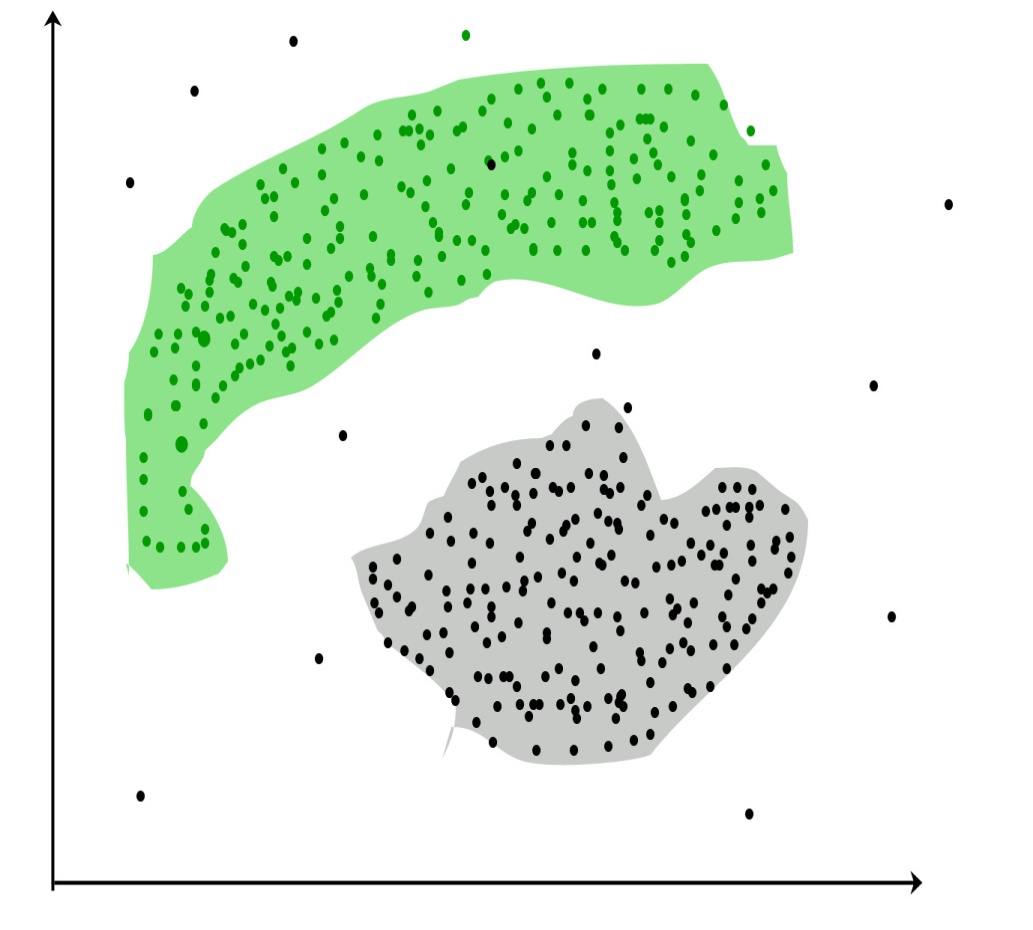
**Types of Regression Models:**  


In [unsupervised learning](https://en.wikipedia.org/wiki/Unsupervised_learning), the algorithm builds a mathematical model of a set of data which contains only inputs and no desired outputs. Unsupervised learning algorithms are used to find structure in the data, like grouping or [clustering](https://en.wikipedia.org/wiki/Cluster_analysis) of data points. Unsupervised learning can discover patterns in the data, and can group the inputs into categories, as in [feature learning](https://en.wikipedia.org/wiki/Feature_learning). [Dimensionality reduction](https://en.wikipedia.org/wiki/Dimensionality_reduction) is the process of reducing the number of "features", or inputs, in a set of data.

 An unsupervised learning method is a method in which we draw references from datasets consisting of input data without labeled responses. Generally, it is used as a process to find meaningful structure, explanatory underlying processes, generative features, and groupings inherent in a set of examples.  
**Clustering** is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups. It is basically a collection of objects on the basis of similarity and dissimilarity between them.

**For ex**– The data points in the graph below clustered together can be classified into one single group. We can distinguish the clusters, and we can identify that there are 3 clusters in the below picture.



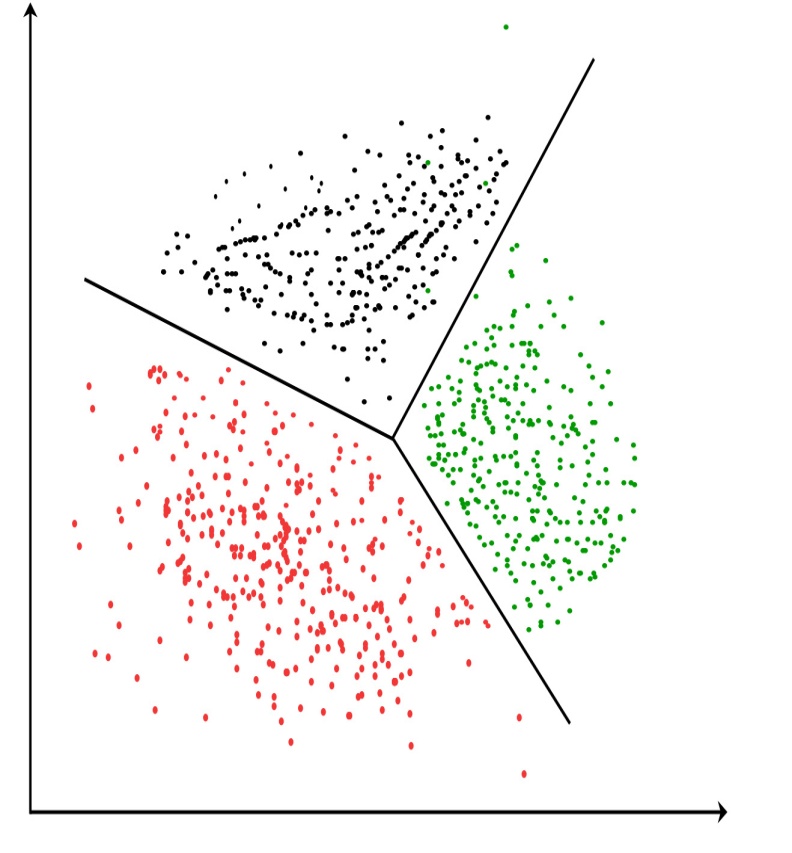
It is not necessary for clusters to be a spherical. Such as :  


These data points are clustered by using the basic concept that the data point lies within the given constraint from the cluster center. Various distance methods and techniques are used for calculation of the outliers.

Clustering is very much important as it determines the intrinsic grouping among the unlabeled data present. There are no criteria for a good clustering. It depends on the user, what is the criteria they may use which satisfy their need. For instance, we could be interested in finding representatives for homogeneous groups (data reduction), in finding “natural clusters” and describe their unknown properties (“natural” data types), in finding useful and suitable groupings (“useful” data classes) or in finding unusual data objects (outlier detection). This algorithm must make some assumptions which constitute the similarity of points and each assumption make different and equally valid clusters.

**Clustering Algorithms :**

[K-means clustering algorithm](https://www.geeksforgeeks.org/k-means-clustering-introduction/) – It is the simplest unsupervised learning algorithm that solves clustering problem.K-means algorithm partition n observations into k clusters where each observation belongs to the cluster with the nearest mean serving as a prototype of the cluster

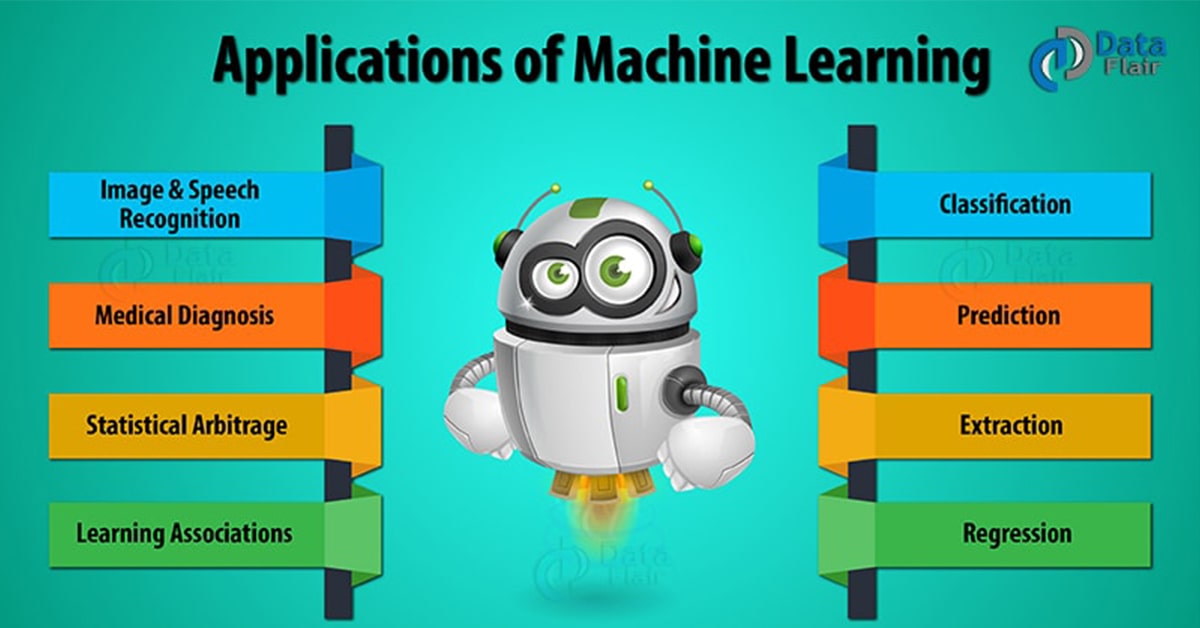
.  


**Applications of Clustering in different fields**

1. **Marketing :** It can be used to characterize & discover customer segments for marketing purposes.  
2. **Biology :** It can be used for classification among different species of plants and animals.  
3. **Libraries :** It is used in clustering different books on the basis of topics and information.  
4. **Insurance :** It is used to acknowledge the customers, their policies and identifying the frauds.  
5. **City Planning :** It is used to make groups of houses and to study their values based on their geographical locations and other factors present.  
6. **Earthquake studies :** By learning the earthquake affected areas we can determine the dangerous zones.

[Active learning](https://en.wikipedia.org/wiki/Active_learning_(machine_learning)) algorithms access the desired outputs (training labels) for a limited set of inputs based on a budget, and optimize the choice of inputs for which it will acquire training labels. When used interactively, these can be presented to a human user for labeling. [Reinforcement learning](https://en.wikipedia.org/wiki/Reinforcement_learning) algorithms are given feedback in the form of positive or negative reinforcement in a dynamic environment, and are used in [autonomous vehicles](https://en.wikipedia.org/wiki/Autonomous_vehicle) or in learning to play a game against a human opponent. Other specialized algorithms in machine learning include [topic modeling](https://en.wikipedia.org/wiki/Topic_modeling), where the computer program is given a set of [natural language](https://en.wikipedia.org/wiki/Natural_language) documents and finds other documents that cover similar topics. Machine learning algorithms can be used to find the unobservable [probability density function](https://en.wikipedia.org/wiki/Probability_density_function) in [density estimation](https://en.wikipedia.org/wiki/Density_estimation) problems. Machine learning seems to be the most straightforward case of all. It is for the most part associated with terms referring to different scientific methods for knowledge discovery or prediction (labelled as machine or statistical learning methods). Towards Data Science provides a platform for thousands of people to exchange ideas and to expand our understanding of data science. Data science is an [interdisciplinary](https://en.wikipedia.org/wiki/Interdisciplinary) field that uses scientific methods, processes, algorithms and systems to extract [knowledge](https://en.wikipedia.org/wiki/Knowledge) and insights from [data](https://en.wikipedia.org/wiki/Data) in various forms, both structured and unstructured similar to [data mining](https://en.wikipedia.org/wiki/Data_mining).

**APPLICATION OF MACHINE LEARNING**



### Machine Learning Applications in Healthcare

### Machine Learning Applications in Finance

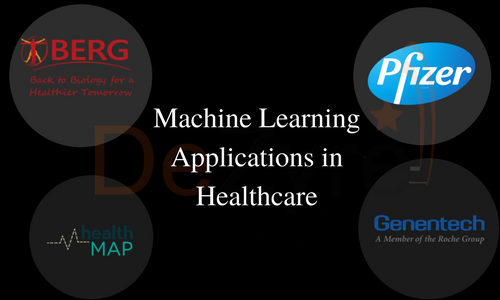
### Machine Learning Applications in Retail

### Machine Learning Applications in Travel

### Machine Learning Applications in Media

**MACHINE LEARNING IN HEALTHCARE**

Doctors and medical practitioners will soon be able to predict with accuracy on how long patients with fatal diseases will live. Medical systems will learn from data and help patients save money by skipping unnecessary tests. Radiologists will be replaced by [machine learning algorithms](https://www.dezyre.com/article/top-10-machine-learning-algorithms/202?utm_source=DeZyre_Blog&utm_medium=TextLink&utm_campaign=Blog_MLApplications_364)



McKinsey Global Institute estimates that applying machine learning techniques to better inform decision making could generate up to $100 billion in value based on optimized innovation, enhanced efficiency of clinical trials and the creation of various novel tools for physicians, insurers and consumers. Computers and Robots cannot replace doctors or nurses, however the use of life-saving technology (machine learning) can definitely transform healthcare domain. When we talk about efficiency of machine learning, more data produces effective results – and the healthcare industry is residing on a data goldmine.

### **i) Drug Discovery/Manufacturing**

Manufacturing or discovering a new drug is expensive and lengthy process as thousands of compounds need to be subjected to a series of tests, and only a single one might result in a usable drug. Machine learning can speed up one or more of these steps in this lengthy multi-step process.

#### **Machine Learning Examples in Healthcare for Drug Discovery**

* Pfizer is using IBM Watson on its immuno-oncology (a technique that uses body’s immune system to help fight cancer) research. This is one of the most significant uses of IBM Watson for drug discovery. Pfizer has been using machine learning for years to sieve through the data to facilitate research in the areas of drug discovery (particularly the combination of multiple drugs) and determine the best participant for a clinical trial.

### **ii) Personalized Treatment/Medication**

Imagine when you walk in to visit your doctor with some kind of an ache in your stomach. After snooping into your symptoms, the doctor inputs them into the computer that extracts the latest research that the doctor might need to know about how to treat your ache. You have an MRI and a computer helps the radiologist detect problems that possibly could be too small for the human eye to see. In the end, a computer scans all your health records and family medical history and compares it to the latest research to advice a treatment protocol that is particularly tailored to your problem. Machine learning is all set to make a mark in personalized care.

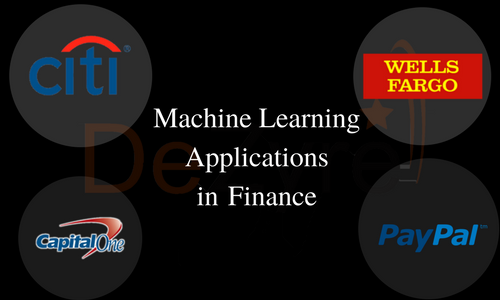
Personalized treatment has great potential for growth in future, and machine learning could play a vital role in finding what kind of genetic makers and genes respond to a particular treatment or medication. Personalized medication or treatment based on individual health records paired with analytics is a hot research area as it provides better disease assessment. In future, increased usage of sensor integrated devices and mobile apps with sophisticated remote monitoring and health-measurement capabilities, there would be another data deluge that could be used for treatment efficacy. Personalized treatment facilitates health optimization and also reduces overall healthcare costs.

#### **Machine Learning Examples in Healthcare for Personalized Treatment**

* A major problem that drug manufacturers often have is that a potential drug sometimes work only on a small group in clinical trial or it could be considered unsafe because a small percentage of people developed serious side effects. Genentech, a member of the Roche Group collaborated with GNS Healthcare to innovate solutions and treatments using biomedical data. Genentech will make use of GNS Reverse Engineering and Forward Simulation to look for patient response markers based on genes which could lead to providing targeted therapies for patients.

## ****MACHINE LEARNING APPLICATIONS IN FINANCE****

More than 90% of the top 50 financial institutions around the world are using machine learning and advanced analytics. The application of machine learning in Finance domain helps banks offer personalized services to customers at lower cost, better compliance and generate greater revenue.



### **Machine Learning Examples in Finance for Fraud Detection**

One of the core machine learning use cases in banking/finance domain is to combat fraud. Machine learning is best suited for this use case as it can scan through huge amounts of transactional data and identify if there is any unusual behavior. Every transaction a customer makes is analyzed in real-time and given a fraud-score that represents the likelihood of the transaction being fraudulent. If the fraud score is above a particular threshold, a rejection will be triggered automatically which would otherwise be difficult without the application of machine learning techniques as humans cannot reviews 1000’s of data points in seconds and make a decision.

* Citibank has collaborated with Portugal based fraud detection company Feedzai that works in real-time to identify and eliminate fraud in online and in-person banking by alerting the customer.
* PayPal is using machine learning to fight money laundering. PayPal has several machine learning tools that compare billions of transactions and can accurately differentiate between what is a legitimate and fraudulent transaction amongst the buyers and sellers.

## **MACHINE LEARNING APPLICATIONS IN RETAIL**

Machine learning in retail is more than just a latest trend, retailers are implementing big data technologies like Hadoop and Spark to build big data solutions and quickly realizing the fact that it’s only the start.



They need a solution which can analyse the data in real-time and provide valuable insights that can translate into tangible outcomes like repeat purchasing. Machine learning algorithms process this data intelligently and automate the analysis to make this supercilious goal possible for retail giants like Amazon, Target, Alibaba and Walmart.

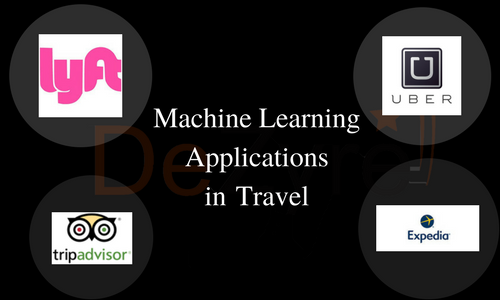
### **Machine Learning Examples in Retail for Product Recommendations**

According to The Realities of Online Personalisation Report, 42% of retailers are using personalized product recommendations using machine learning technology. It is no secret that customers always look for personalized shopping experiences, and these recommendations increase the conversion rates for the retailers resulting in fantastic revenue.

* The moment you start browsing for items on Amazon, you see recommendations for products you are interested in as “Customers Who Bought this Product Also Bought” and “Customers who viewed this product also viewed”, as well specific tailored product recommendation on the home page, and through email. Amazon uses Artificial Neural Networks machine learning algorithm to generate these recommendations for you.
* To make smart personalized recommendations, Alibaba has developed “E-commerce Brain” that makes use of real-time online data to build machine learning models for predicting what customers want and recommending the relevant products based on their recent order history, bookmarking, commenting, browsing history,  and other actions.

## **MACHINE LEARNING APPLICATIONS IN TRAVEL**

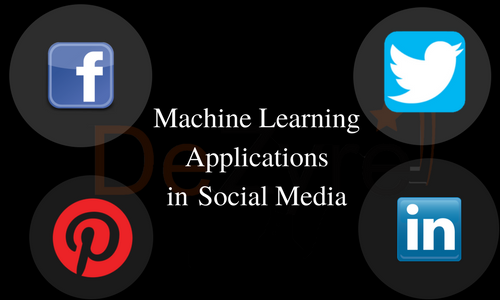
One of Uber’s biggest uses of machine learning comes in the form of surge pricing, a machine learning model nicknamed as “Geosurge” at Uber. If you are getting late for a meeting and you need to book an Uber in crowded area, get ready to pay twice the normal fare. In 2011, during New Year’s Eve in New York, Uber charged $37 to $135 for one mile journey



. Uber leverages predictive modelling in real-time based on traffic patterns, supply and demand. Uber has acquired a patent on surge pricing. However, customer backlash on surge-pricing is strong, so Uber is using machine learning to predict where demand will be high so that drivers can prepare in advance to meet the demand, and surge pricing can be reduced to a greater extent.

## **MACHINE LEARNING APPLICATIONS IN SOCIAL MEDIA**

Machine learning offers the most efficient means of engaging billions of social media users. From personalizing news feed to rendering targeted ads, machine learning is the heart of all social media platforms for their own and user benefits. Social media and chat applications have advanced to a great extent that users do not pick up the phone or use email to communicate with brands – they leave a comment on Facebook or Instagram expecting a speedy reply than the traditional channels



.

Here are some machine learning examples that you must be using and loving in your social media accounts without knowing the fact that there interesting features are machine learning applications -

* Earlier Facebook used to prompt users to tag your friends but nowadays the social networks artificial neural networks machine learning algorithm identifies familiar faces from contact list. The ANN algorithm mimics the structure of human brain to power facial recognition.
* The professional network LinkedIn knows where you should apply for your next job, whom you should connect with and how your skills stack up against your peers as you search for new job.

**PRINCIPAL COMPONENT ANALYSIS**

The main idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent. The same is done by transforming the variables to a new set of variables, which are known as the principal components (or simply, the PCs) and are orthogonal, ordered such that the retention of variation present in the original variables decreases as we move down in the order. So, in this way, the 1st principal component retains maximum variation that was present in the original components. The principal components are the eigenvectors of a covariance matrix, and hence they are orthogonal.

Importantly, the dataset on which PCA technique is to be used must be scaled. The results are also sensitive to the relative scaling. As a layman, it is a method of summarizing data. Imagine some wine bottles on a dining table. Each wine is described by its attributes like colour, strength, age, etc. But redundancy will arise because many of them will measure related properties. So what PCA will do in this case is summarize each wine in the stock with less characteristics.

Intuitively, Principal Component Analysis can supply the user with a lower-dimensional picture, a projection or "shadow" of this object when viewed from its most informative viewpoint.

Data science is a "concept to unify statistics, data analysis, machine learning and their related methods" in order to "understand and analyze actual phenomena" with data It employs techniques and theories drawn from many fields within the context of [mathematics](https://en.wikipedia.org/wiki/Mathematics), [statistics](https://en.wikipedia.org/wiki/Statistics), [information science](https://en.wikipedia.org/wiki/Information_science), and [computer science](https://en.wikipedia.org/wiki/Computer_science). Data analysis is a process of inspecting, [cleansing](https://en.wikipedia.org/wiki/Data_cleansing), [transforming](https://en.wikipedia.org/wiki/Data_transformation), and [modeling](https://en.wikipedia.org/wiki/Data_modeling) [data](https://en.wikipedia.org/wiki/Data) with the goal of discovering useful information, informing conclusions, and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, while being used in different business, science, and social science domains. In today's business, data analysis is playing a role in making decisions more scientific and helping the business achieve effective operation.

**Existing System**

In previous, The project is based on support vector method, the k-nearest neighbor method, and the random forest method.

**Disadvantages**

* SVM algorithm is not suitable for large data sets. SVM does not perform very well when the data set has more noise i.e. target classes are overlapping. In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.
* Does not work well with large dataset as calculating distances between each data instance would be very costly and does not work well with high dimensionality as this will complicate the distance calculating process to calculate distance for each dimension.Sensitive to noisy and missing data.
* random forests can be an improvement on single decision trees, more sophisticated techniques are available. Prediction accuracy on complex problems is usually inferior to gradient-boosted trees. A forest is less interpretable than a single decision tree.

**Proposed System**

This project describes an experiment that results in a model for detecting XSS attacks. Machine learning algorithms are considered: the method Multinomial Bayesian classifier. Multinomial Naive Bayes (MNB) is a popular classification algorithm that is commonly used in natural language processing (NLP) tasks such as text classification and sentiment analysis.The algorithm is based on the Bayes theorem which calculates the probability of a certain event occurring given the probabilities of related events. In the case of MNB, the probabilities of different words occurring in a document are used to calculate the probability of the document belonging to a certain class.

**Advantages**

* It can be implemented quickly with minimal computational resources.
* Fast and efficient: MNB is computationally efficient and can handle large datasets with many features.
* Handles large number of features: MNB can handle a large number of features, which makes it particularly useful for text classification tasks, where the input data consists of a large number of words.

**Module description:-**

* + - Data collection
    - Data Pre-Processing
    - Training data and Test data
    - Model Creation
    - Model Prediction
* **Data collection**

To get a model with good accuracy, you need to assemble and prepare the source dataset correctly. The data set must contain malicious and benign JavaScript code. The data set for the experiment consists of 240,000 instances of malicious and benign code. Malicious code is 40,000 instance [7], and benign code is 200,000 instance

* **Data Pre-processing**

The source data set is represented as queries with a different number of parameters. Each request is decoded into Unicode characters. After decoding, parameters of this query are extracted using regular expressions. Abnormal queries containing a large number of parameters are removed from the dataset.

* **Training data and Test data** 
  + - For choosing a model we split our dataset into train and test
    - Here data’s are split into 3:1 ratio that means
    - Training data having 70 percent and testing data having 30 percent
    - In this split process preforming based on train\_test\_split model
    - After splitting we get xtrain xtest and ytrain ytest
* **Model Creation** 
  + Contextualise machine learning in your organisation.
  + Explore the data and choose the type of algorithm.
  + Prepare and clean the dataset.
  + Split the prepared dataset and perform cross validation.
  + Perform machine learning optimisation.
  + Deploy the model
* **Model Prediction**

Predictive modeling is a statistical technique using machine learning and data mining to predict and forecast likely future outcomes with the aid of historical and existing data. It works by analyzing current and historical data and projecting what it learns on a model generated to forecast likely outcomes. In this Project, our final prediction is to predict whether a script should be safe or malicious.

**Algorithm Implementation**

**Multinomial Naïve Bayes Algorithm:-**

The Multinomial Naive Bayes algorithm is a Bayesian learning approach popular in Natural Language Processing (NLP). The program guesses the tag of a text, such as an email or a newspaper story, using the Bayes theorem. It calculates each tag's likelihood for a given sample and outputs the tag with the greatest chance.

**Introduction**

With an ever-growing amount of textual information stored in electronic form such as legal documents, policies, company strategies, etc., automatic text classification is becoming increasingly important. This requires a supervised learning technique that classifies every new document by assigning one or more class labels from a fixed or predefined class. It uses the bag of words approach, where the individual words in the document constitute its features, and the order of the words is ignored. This technique is different from the way we communicate with each other. It treats the language like it’s just a bag full of words and each message is a random handful of them. Large documents have a lot of words that are generally characterized by very high dimensionality feature space with thousands of features. Hence, the learning algorithm requires to tackle high dimensional problems, both in terms of classification performance and computational speed.

Naïve Bayes, which is computationally very efficient and easy to implement, is a learning algorithm frequently used in text classification problems. Two event models are commonly used:

Multivariate Bernoulli Event Model

Multivariate Event Model

The Multivariate Event model is referred to as Multinomial Naive Bayes.

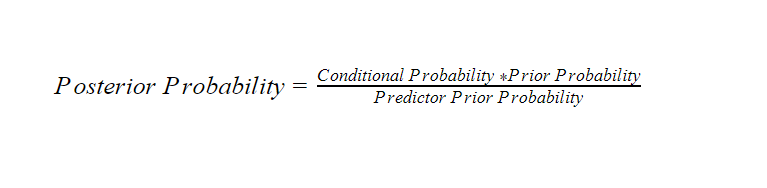
When most people want to learn about Naive Bayes, they want to learn about the Multinomial Naive Bayes Classifier. However, there is another commonly used version of Naïve Bayes, called Gaussian Naive Bayes Classification. Check out the free course on naive Bayes supervised learning.

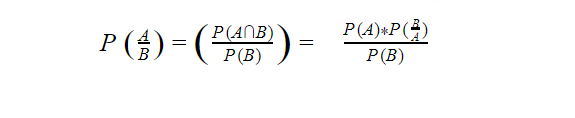
Naive Bayes is based on Bayes’ theorem, where the adjective Naïve says that features in the dataset are mutually independent. Occurrence of one feature does not affect the probability of occurrence of the other feature. For small sample sizes, Naïve Bayes can outperform the most powerful alternatives. Being relatively robust, easy to implement, fast, and accurate, it is used in many different fields. Check out naïve bayes classifiers.

For Example, Spam filtering in email, Diagnosis of diseases, making decisions about treatment, Classification of RNA sequences in taxonomic studies, to name a few. However, we have to keep in mind about the type of data and the type of problem to be solved that dictates which classification model we want to choose. Strong violations of the independence assumptions and non-classification problems can lead to poor performance. In practice, it is recommended to use different classification models on the same dataset and then consider the performance, as well as computational efficiency.

Also Read: Top Machine Learning Interview Questions

To understand how Naïve Bayes works, first, we have to understand the concept of Bayes’ rule. This probability model was formulated by Thomas Bayes (1701-1761) and can be written as:





where,

PA= the prior probability of occurring A

PBA= the condition probability of B given that A occurs

PAB= the condition probability of A given that B occurs

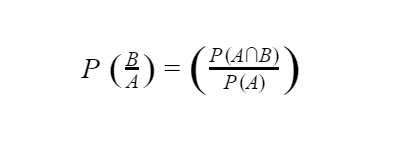
PB= the probability of occuring B

The posterior probability, can be interpreted as: What is the revised probability of an event occurring after taking new information into consideration?

It is a better reflection of the underlying truth of a data generating process because it includes more information.

Conditional Probability

The probability of one event A occurring when another event B with some relationship to A has already occurred is called conditional probability.



This expression is valid only when P(A) is greater than zero.

Prior Probability

This probability can be defined as the prior knowledge or belief i.e. the probability of an event computed before the collection of new data. This probability is revised as new information becomes available to produce more accurate results.

If the prior observations are used to calculate the probability, we call it prior probability.

The Bag of Words Model

Feature extraction and Selection are the most important sub-tasks in pattern classification. The three main criteria of good features are:

Salient: The features should be meaningful and important to the problem

Invariant: The features are resistant to scaling, distortion and orientation etc.

Discriminatory: For training of classifiers, the features should have enough information to distinguish between patterns.

Bag of words is a commonly used model in Natural Language Processing. The idea behind this model is the creation of vocabulary that contains the collection of different words, and each word is associated with a count of how it occurs. Later, the vocabulary is used to create d-dimensional feature vectors.

For Example:

D1: Each country has its own constitution

D2: Every country has its own uniqueness

Vocabulary could be written as:

V= {each : 1, state : 1, has : 2, its : 2, own : 2, constitution : 1, every : 1, country : 2,}

Tokenization

It is the process of breaking down the text corpus into individual elements. These individual elements act as an input to machine learning algorithms.

For Example:

Every country has its own uniqueness



Stop Words

Stop Words also known as un-informative words such as (so, and, or, the) should be removed from the document.

Stemming and Lemmatization

Stemming and Lemmatization are the process of transforming a word into its root form and aims to obtain the grammatically correct forms of words.

The above-mentioned process comes under the Bag of Words Model. Multinomial Naïve Bayes uses term frequency i.e. the number of times a given term appears in a document. Term frequency is often normalized by dividing the raw term frequency by the document length. After normalization, term frequency can be used to compute maximum likelihood estimates based on the training data to estimate the conditional probability.

Example:

Let me explain a Multinomial Naïve Bayes Classifier where we want to filter out the spam messages. Initially, we consider eight normal messages and four spam messages. Now, imagine we received a lot of emails from friends, family, office and we also received spam (unwanted messages that are usually scams or unsolicited advertisements).

Let see the histogram of all the words that occur in the normal messages from family and friends.

We can use the histogram to calculate the probabilities of seeing each word, given that it was a normal message. The probability of word dear given that we saw in normal message is-

Probability (Dear|Normal) = 8 /17 = 0.47

Similarly, the probability of word Friend is-

Probability (Friend/Normal) = 5/ 17 =0.29

Probability (Lunch/Normal) = 3/ 17 =0.18

Probability (Money/Normal) = 1/ 17 =0.06

Now let’s make the histogram of all the words in spam.

The probability of word dear given that we saw in spam message is-

Probability (Dear|Spam) = 2 /7 = 0.29

Similarly, the probability of word Friend is-

Probability (Friend/Spam) = 1/ 7 =0.14

Probability (Lunch/Spam) = 0/ 7 =0.00

Probability (Money/Spam) = 4/ 7 =0.57

Here, we have calculated the probabilities of discrete words and not the probability of something continuous like weight or height. These Probabilities are also called Likelihoods.

Now, let’s say we have received a normal message as Dear Friend and we want to find out if it’s a normal message or spam.

We start with an initial guess that any message is a Normal Message.

From our initial assumptions of 8 Normal messages and 4 Spam messages, 8 out of 12 messages are normal messages. The prior probability, in this case, will be:

Probability (Normal) = 8 / (8+4) = 0.67

We multiply this prior probability with the probabilities of Dear Friend that we have calculated earlier.

0.67 \* 0.47 \* 0.29 = 0.09

0.09 is the probability score considering Dear Friend is a normal message.

Alternatively, let’s say that any message is a Spam.

4 out of 12 messages are Spam. The prior probability in this case will be:

Probability (Normal) = 4 / (8+4) = 0.33

Now we multiply the prior probability with the probabilities of Dear Friend that we have calculated earlier.

0.33 \* 0.29 \* 0.14 = 0.01

0.01 is the probability score considering Dear Friend is a Spam.

The probability score of Dear Friend being a normal message is greater than the probability score of Dear Friend being spam. We can conclude that Dear Friend is a normal message.

Naive Bayes treats all words equally regardless of how they are placed because it’s difficult to keep track of every single reasonable phrase in a language.

Difference between Bernoulli, Multinomial and Gaussian Naive Bayes

Multinomial Naïve Bayes consider a feature vector where a given term represents the number of times it appears or very often i.e. frequency. On the other hand, Bernoulli is a binary algorithm used when the feature is present or not. At last Gaussian is based on continuous distribution.

**CountVectorizer**

CountVectorizer is a machine learning tool used for natural language processing and text classification tasks. It is a commonly used tool in the context of web application security, specifically in cross-site scripting (XSS) detection and prevention.

XSS is a type of attack where an attacker injects malicious scripts into a web page viewed by unsuspecting users. These scripts can be used to steal sensitive information or perform other malicious actions. To prevent XSS attacks, web applications can use input validation and output encoding to filter out malicious scripts. CountVectorizer can be used in this context to automatically identify and categorize potentially malicious scripts by analyzing the text content of the web page.

Here's how CountVectorizer can be used for XSS detection:

Collect a set of web pages, some of which are known to be vulnerable to XSS attacks and some of which are not.

Preprocess and clean the text data from the web pages by removing stop words, non-alphabetic characters, and converting all text to lowercase.

Initialize the CountVectorizer and fit it to the preprocessed text data.

Transform the text data into a matrix of word counts using the CountVectorizer's transform method.

Train a machine learning classifier (such as a logistic regression model) on the word frequency matrix and the labels (vulnerable or not vulnerable).

Use the trained classifier to predict the vulnerability of new web pages by transforming their text data into a word frequency matrix and applying the classifier's predict method.

By using CountVectorizer in conjunction with a machine learning classifier, web applications can automatically detect and prevent XSS attacks without relying solely on manual input validation and output encoding. However, it's important to note that machine learning models can never be 100% accurate, and attackers can always find new and creative ways to bypass detection techniques. Therefore, it's important to use CountVectorizer as one part of a multi-layered approach to web application security.

**Block Diagram**

OUTPUT

TEST DATA

TEST DATA

MODEL PREDICTION

MODEL CREATION

DATA TRAINING PROCESS

TRAIN DATA

DATA PRE-PROCESSING

Dataset

**Literature Review**

**Title : Machine Learning Based Cross-Site Scripting Detection in Online Social Network**

**Author : Rui Wang; Xiaoqi Jia; Qinlei Li; Shengzhi Zhang**

**Abstract:**

Nowadays online social network (OSN) is one of the most popular internet services in the world. It allows us to communicate with others and share knowledge. However, from the security's point of view, OSN is becoming the favorite target for the attackers, and is under a lot of threats such as cross-site scripting (XSS) attacks. In this paper, we present a novel approach using machine learning to do XSS detection in OSN. Firstly, we leverage a new method to capture identified features from web pages and then establish classification models which can be used in XSS detection. Secondly, we propose a novel method to simulate XSS worm spreading and build our webpage database. Finally, we set up experiments to verify the classification models using our test database. Our experiment results demonstrate that our approach is an effective countermeasure to detect the XSS attack. **Literature Review**

**Title : Classification of XSS Attacks by Machine Learning with Frequency of Appearance and Co-occurrence**

**Author : Sota Akaishi; Ryuya Uda**

**Abstract:**

**Abstract:**

Nowadays, Sentiment Analysis has become an active research area due to the availability of many opinionated data through increased activity in Blogging, Tagging, Podcasting, social networking sites, RSS feeds, and Social Bookmarking. In the present situation, the whole world is facing the crisis of the COVID-19 pandemic. Particularly, let's talk about nationwide lockdown in India to control the spread of COVID-19. The government relies on social media to observe people's aviews on their policies during the lockdown. In this paper, Twitter data has been used for Sentiment Analysis, which focus on people opinion during the COVID-19 nationwide Lockdown effect in India. Different keywords data was collected on various dates between March 25, 2020, to June 09, 2020. This research work is an application of the real-time TextBlob sentiment analyzer tool built based on the Natural Language Toolkit (NLTK). Relevant keyword tweets were extracted by tweeter API. Then a model was trained to classify the result on a specific opinion. This NLPbased sentiment analysis model is ideal for analyzing the emotions while tested with seven primary keywords: Lockdown1.0, Migrant Workers, Indian Economic, ICMR, Lockdown5.0, Medical Facilities, and Police. The result shows that Lockdown 1.0 got the most positive sentiments, followed by ICMR and Medical Facility.

**Literature Review**

**Title : Sentiment Analysis of Lockdown in India During COVID-19: A Case Study on Twitter**

**Author : Prasoon Gupta; Sanjay Kumar; R. R. Suman; Vinay Kumar**

**Abstract:**

With the rapid increase in the use of the Internet, sentiment analysis has become one of the most popular fields of natural language processing (NLP). Using sentiment analysis, the implied emotion in the text can be mined effectively for different occasions. People are using social media to receive and communicate different types of information on a massive scale during COVID-19 outburst. Mining such content to evaluate people's sentiments can play a critical role in making decisions to keep the situation under control. The objective of this study is to mine the sentiments of Indian citizens regarding the nationwide lockdown enforced by the Indian government to reduce the rate of spreading of Coronavirus. In this work, the sentiment analysis of tweets posted by Indian citizens has been performed using NLP and machine learning classifiers. From April 5, 2020 to April 17, 2020, a total of 12 741 tweets having the keywords “Indialockdown” are extracted. Data have been extracted from Twitter using Tweepy API, annotated using TextBlob and VADER lexicons, and preprocessed using the natural language tool kit provided by the Python. Eight different classifiers have been used to classify the data. The experiment achieved the highest accuracy of 84.4% with LinearSVC classifier and unigrams. This study concludes that the majority of Indian citizens are supporting the decision of the lockdown implemented by the Indian government during corona outburst.

**System Requirements**

* + Windows 10
  + Ram : 4GB or 8GB
  + Processor : i3 or i5
  + Python 3.9

# **SOFTWARE DESCRIPTION**

**PYTHON:**

Python is an open source programming language. Python was made to be easy-to-read and powerful. A Dutch programmer named Guido van Rossum made Python in 1991. He named it after the television show Monty Python's Flying Circus. Many Python examples and tutorials include jokes from the show.

Python is an interpreted language. Interpreted languages do not need to be compiled to run. A program called an interpreter runs Python code on almost any kind of computer. This means that a programmer can change the code and quickly see the results. This also means Python is slower than a compiled language like C, because it is not running machine code directly.

Python is a good programming language for beginners. It is a high-level language, which means a programmer can focus on what to do instead of how to do it. Writing programs in Python takes less time than in some other languages.

Python drew inspiration from other programming languages like C, C++, Java, Perl, and Lisp.

Python has a very easy-to-read syntax. Some of Python's syntax comes from C, because that is the language that Python was written in. But Python uses whitespace to delimit code: spaces or tabs are used to organize code into groups. This is different from C. In C, there is a semicolon at the end of each line and curly braces ({}) are used to group code. Using whitespace to delimit code makes Python a very easy-to-read language.

Python is used by hundreds of thousands of programmers and is used in many places. Sometimes only Python code is used for a program, but most of the time it is used to do simple jobs while another programming language is used to do more complicated tasks.

Its standard library is made up of many functions that come with Python when it is installed. On the Internet there are many other libraries available that make it possible for the Python language to do more things. These libraries make it a powerful language; it can do many different things.

Some things that Python is often used for are:

Web development

Game programming

Desktop GUIs

Scientific programming

Network programming.

**Version 1**

Python reached version 1.0 in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce. Van Rossum stated that "Python acquired lambda, reduce(), filter() and map(), courtesy of a Lisp hacker who missed them and submitted working patches’’

The last version released while Van Rossum was at CWI was Python 1.2. In 1995, Van Rossum continued his work on Python at the Corporation for National Research Initiatives (CNRI) in Reston, Virginia whence he released several versions.

By version 1.4, Python had acquired several new features. Notable among these are the Modula-3 inspired keyword arguments (which are also similar to Common Lisp's keyword arguments) and built-in support for complex numbers. Also included is a basic form of data hiding by name mangling, though this is easily bypassed.

During Van Rossum's stay at CNRI, he launched the Computer Programming for Everybody (CP4E) initiative, intending to make programming more accessible to more people, with a basic "literacy" in programming languages, similar to the basic English literacy and mathematics skills required by most employers. Python served a central role in this: because of its focus on clean syntax, it was already suitable, and CP4E's goals bore similarities to its predecessor, ABC. The project was funded by DARPA.[13] As of 2007, the CP4E project is inactive, and while Python attempts to be easily learnable and not too arcane in its syntax and semantics, reaching out to non-programmers is not an active concern.

In 2000, the Python core development team moved to BeOpen.com to form the BeOpen Python Labs team. CNRI requested that a version 1.6 be released, summarizing Python's development up to the point at which the development team left CNRI. Consequently, the release schedules for 1.6 and 2.0 had a significant amount of overlap.Python 2.0 was the only release from BeOpen.com. After Python 2.0 was released by BeOpen.com, Guido van Rossum and the other PythonLabs developers joined Digital Creations.

The Python 1.6 release included a new CNRI license that was substantially longer than the CWI license that had been used for earlier releases. The new license included a clause stating that the license was governed by the laws of the State of Virginia. The Free Software Foundation argued that the choice-of-law clause was incompatible with the GNU General Public License. BeOpen, CNRI and the FSF negotiated a change to Python's free software license that would make it GPL-compatible. Python 1.6.1 is essentially the same as Python 1.6, with a few minor bug fixes, and with the new GPL-compatible license.

**Version 2**

Python 2.0 introduced list comprehensions, a feature borrowed from the functional programming languages SETL and Haskell. Python's syntax for this construct is very similar to Haskell's, apart from Haskell's preference for punctuation characters and Python's preference for alphabetic keywords. Python 2.0 also introduced a garbage collection system capable of collecting reference cycles.[7]

Python 2.1 was close to Python 1.6.1, as well as Python 2.0. Its license was renamed Python Software Foundation License. All code, documentation and specifications added, from the time of Python 2.1's alpha release on, is owned by the Python Software Foundation (PSF), a non-profit organization formed in 2001, modeled after the Apache Software Foundation.[15] The release included a change to the language specification to support nested scopes, like other statically scoped languages.[16] (The feature was turned off by default, and not required, until Python 2.2.)

A major innovation in Python 2.2 was the unification of Python's types (types written in C) and classes (types written in Python) into one hierarchy. This single unification made Python's object model purely and consistently object oriented. Also added were generators which were inspired by Icon.

Python 2.5 was released on September 2006 and introduced the with statement, which encloses a code block within a context manager (for example, acquiring a lock before the block of code is run and releasing the lock afterwards, or opening a file and then closing it), allowing Resource Acquisition Is Initialization (RAII)-like behavior and replacing a common try/finally idiom.

Python 2.6 was released to coincide with Python 3.0, and included some features from that release, as well as a "warnings" mode that highlighted the use of features that were removed in Python 3.0.Similarly, Python 2.7 coincided with and included features from Python 3.1,which was released on June 26, 2009.

Parallel 2.x and 3.x releases then ceased, and Python 2.7 was the last release in the 2.x series. In November 2014, it was announced that Python 2.7 would be supported until 2020, but users were encouraged to move to Python 3 as soon as possible.

**Version 3**

Python 3.0 (also called "Python 3000" or "Py3K") was released on December 3, 2008 It was designed to rectify fundamental design flaws in the language—the changes required could not be implemented while retaining full backwards compatibility with the 2.x series, which necessitated a new major version number. The guiding principle of Python 3 was: "reduce feature duplication by removing old ways of doing things".

Python 3.0 was developed with the same philosophy as in prior versions. However, as Python had accumulated new and redundant ways to program the same task, Python 3.0 had an emphasis on removing duplicative constructs and modules, in keeping with "There should be one— and preferably only one —obvious way to do it".

Nonetheless, Python 3.0 remained a multi-paradigm language. Coders still had options among object-orientation, structured programming, functional programming and other paradigms, but within such broad choices, the details were intended to be more obvious in Python 3.0 than they were in Python 2.x.

**Python 2.7.0**

Note: A bugfix release, 2.7.13, is currently [available](https://www.python.org/downloads/release/python-2713/). Its use is recommended.

Python 2.7.0 was released on July 3rd, 2010.

Python 2.7 is scheduled to be the last major version in the 2.x series before it moves into an extended maintenance period. This release contains many of the features that were first released in Python 3.1. Improvements in this release include:

* An ordered dictionary type
* New unittest features including test skipping, new assert methods, and test discovery
* A much faster io module
* Automatic numbering of fields in the str.format() method
* Float repr improvements backported from 3.x
* Tile support for Tkinter
* A backport of the memoryview object from 3.x
* Set literals
* Set and dictionary comprehensions
* Dictionary views
* New syntax for nested with statements

MODULES

1.NUMPY

2.PANDAS

3.SKLEARN

4.FLASK

NUMPY:-

NumPy is a Python package. It stands for &#39;Numerical Python&#39;. It is a library

consisting of multidimensional array objects and a collection of routines for

processing of array.

Numeric, the ancestor of NumPy, was developed by Jim Hugunin. Another

package Numarray was also developed, having some additional functionalities. In

2005, Travis Oliphant created NumPy package by incorporating the features of

Numarray into Numeric package. There are many contributors to this open source

project.

Operations using NumPy

Using NumPy, a developer can perform the following operations −

 Mathematical and logical operations on arrays.

 Fourier transforms and routines for shape manipulation.

 Operations related to linear algebra. NumPy has in-built functions for linear

algebra and random number generation.

NumPy – A Replacement for MatLab

NumPy is often used along with packages like SciPy (Scientific Python)

and Mat−plotlib (plotting library). This combination is widely used as a

replacement for MatLab, a popular platform for technical computing. However,

Python alternative to MatLab is now seen as a more modern and complete

programming language.

It is open source, which is an added advantage of NumPy.

INSTALLATION

pip install &quot;numpy”

**PANDAS**

Pandas is an open-source, BSD-licensed Python library providing high-

performance, easy-to-use data structures and data analysis tools for the Python

programming language. Python with Pandas is used in a wide range of fields

including academic and commercial domains including finance, economics,

Statistics, analytics, etc. In this tutorial, we will learn the various features of

Python Pandas and how to use them in practice.

Pandas deals with the following three data structures −

* Series
* DataFrame
* Panel

These data structures are built on top of Numpy array, which means they are fast.

INSTALLATION

Pip install pandas

SKLEARN

Scikit-learn is a machine learning library for Python. It features several regression,

classification and clustering algorithms including SVMs, gradient boosting, k-

means, random forests and DBSCAN. It is designed to work with

Python  Numpy  and  SciPy .

The scikit-learn project kicked off as a Google Summer of Code (also known as

GSoC) project by David Cournapeau as scikits.learn. It gets its name from

“Scikit”, a separate third-party extension to SciPy.

Python Scikit-learn

Scikit is written in Python (most of it) and some of its core algorithms are written

in Cython for even better performance.

Scikit-learn is used to build models and it is not recommended to use it for reading,

manipulating and summarizing data as there are better frameworks available for

the purpose.

It is open source and released under BSD license.

Install Scikit Learn

Scikit assumes you have a running Python 2.7 or above platform with NumPY

(1.8.2 and above) and SciPY (0.13.3 and above) packages on your device. Once we

have these packages installed we can proceed with the installation.

pip install scikit-learn

Flask:

What is Flask?

Flask is an API of Python that allows us to build up web-applications. It was

developed by Armin Ronacher. Flask’s framework is more explicit than Django’s

framework and is also easier to learn because it has less base code to implement a

simple web-Application. A Web-Application Framework or Web Framework is

the collection of modules and libraries that helps the developer to write

applications without writing the low-level codes such as protocols, thread

management, etc. Flask is based on WSGI(Web Server Gateway Interface) toolkit

and Jinja2 template engine.

Routing:

Nowadays, the web frameworks provide routing technique so that user can

remember the URLs. It is useful to access the web page directly without

navigating from the Home page. It is done through the

following route() decorator, to bind the URL to a function.

Building URL in FLask:

Dynamic Building of the URL for a specific function is done

using url\_for() function. The function accepts the name of the function as first

argument, and one or more keyword arguments. See this example.

**Conclusion**

Cross-site scripting (XSS) is a significant security issue for web applications, and machine learning techniques can be used to detect and prevent such attacks. Multinomial Naive Bayes (MNB) with Count Vectorizer is one such approach that can be used for XSS detection.

Count Vectorizer is a feature extraction technique that transforms text data into numerical vectors, which can then be used as input to machine learning algorithms. MNB with Count Vectorizer uses the frequency counts of words in the text as features, and the algorithm can be trained on a dataset of known malicious and benign web pages to classify new web pages as either malicious or benign.

The advantage of using MNB with Count Vectorizer for XSS detection is that it is computationally efficient and can handle large datasets with many features. It also performs well with high-dimensional data and can identify important features that contribute to the classification task. Additionally, this approach can be used to analyze the content of web pages without requiring knowledge of the underlying web application code.

However, the performance of MNB with Count Vectorizer for XSS detection may be affected by the quality and size of the training dataset and the effectiveness of the feature extraction technique. In addition, this approach may not be able to detect more sophisticated XSS attacks that use advanced evasion techniques.

In conclusion, MNB with Count Vectorizer is a promising approach for XSS detection, and it can provide efficient and effective protection for web applications. However, it is important to consider the limitations of this approach and to continue research to improve its performance and address its weaknesses.

**Reference**

* S. Kirsten, “Cross Site Scripting (XSS),” OWASP Foundation, 2020.
* B. Glas, “OWASP Top 10 Security Risks & Vulnerabilities,” OWASP Foundation, 2020.
* A. Johari and S. K. Adnan, “Defending Malicious Script Attacks Using Machine Learning Classifiers,” Wireless Communications and Mobile Computing, 2017.
* F. A. Mereani and J. M. Howe, “Detecting Cross-Site Scripting Attacks Using Machine Learning,” Advanced Machine Learning Technologies and Applications, vol. 723, pp. 200–210, 2018.
* F. A. Mereani and J. M. Howe, “Preventing Cross-Site Scripting Attacks by Combining Classifiers,” Proc. of the 10th Int. Joint Conf. on Computational Intelligence, vol. 1, pp. 135–143, 2018.
* F. A. Mereani and J. M. Howe, “Exact and approximate rule extraction from neural networks with boolean features,” Proc. of the 11th Int. Joint Conf. on Computational Intelligence, vol. 1, pp 424–433, 2019.
* XSS Archive, In Famous and Government Websites.
* S. S. Shah, Cross site scripting XSS dataset for Deep learning. Kaggle, 2020.
* Word2vec, The Python Package Index. [Online]. Available: https://pypi.org/project/word2vec/
* TSNE, scikit-learn: machine learning in Python. [Online]. Available: https://scikitlearn.org/stable/modules/generated/sklearn.manifold.TSNE.html