

from a dialogue based clinical expert system, etc.

The field of NLP involves making computers to perform useful tasks with the natural language. The input and output of an NLP system can be –

•Speech

•Written Text

Adjective

Adverb

Components of NLP

There are two components of NLP as given –

Natural Language Understanding (NLU)

Understanding involves the following tasks –

- Mapping the given input in natural language into useful representations.
- Analyzing different aspects of the language.

Natural Language Generation (NLG)

It is the process of producing meaningful phrases and sentences in the form of natural language from some internal representation.

It involves –

- Text planning – It includes retrieving the relevant content from knowledge base.
- Sentence planning – It includes choosing required words, forming meaningful phrases, setting tone of the sentence.
- Text Realization – It is mapping sentence plan into sentence structure.

The NLU is harder than NLG.

Syntax and semantic analysis are two main techniques used with natural language processing. Syntax is the arrangement of words in a sentence to make grammatical sense. NLP uses syntax to assess meaning from a language based on grammatical rules. Syntax techniques used include parsing (grammatical analysis for a sentence), word segmentation (which divides a large piece of text to units), sentence breaking (which places sentence boundaries in large texts),

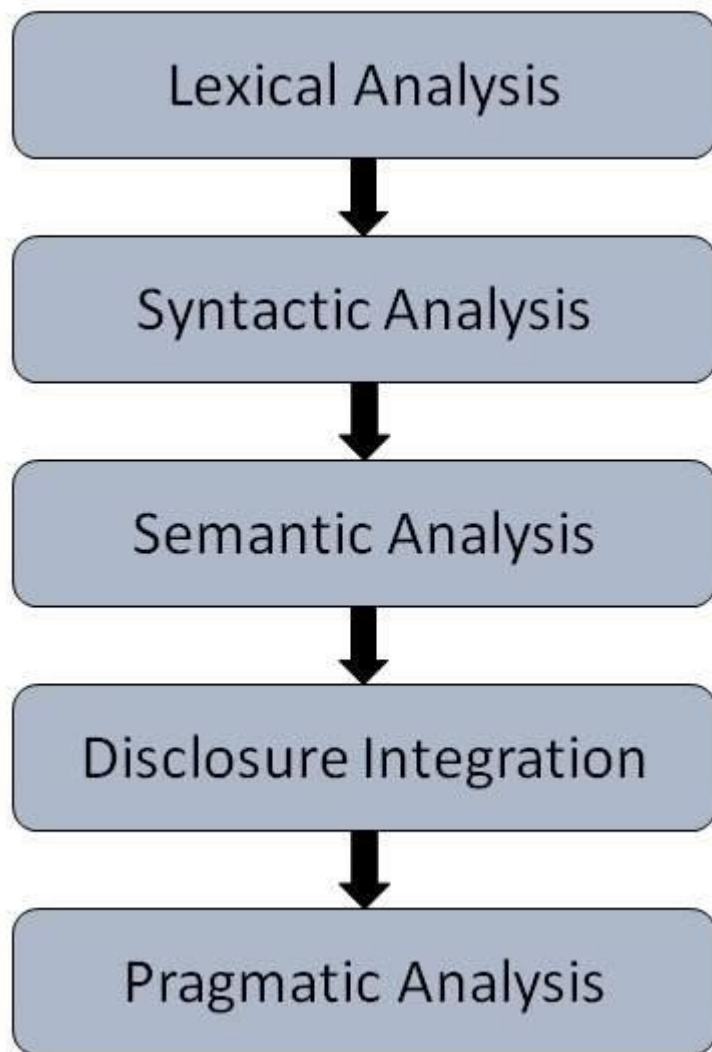
morphological segmentation (which divides words into groups) and stemming (which divides words with inflection in them to root forms).

Semantics involves the use and meaning behind words. NLP applies algorithms to understand the meaning and structure of sentences. Techniques that NLP uses with semantics include word sense disambiguation (which derives meaning of a word based on context), named entity recognition (which determines words that can be categorized into groups), and natural language generation (which will use a database to determine semantics behind words).

Steps in NLP

There are general five steps –

- Lexical Analysis – It involves identifying and analyzing the structure of words. Lexicon of a language means the collection of words and phrases in a language. Lexical analysis is dividing the whole chunk of txt into paragraphs, sentences, and words.**
- Syntactic Analysis (Parsing) – It involves analysis of words in the sentence for grammar and arranging words in a manner that shows the relationship among the words. The sentence such as “The school goes to boy” is rejected by English syntactic analyzer.**



•**Semantic Analysis** – It draws the exact meaning or the dictionary meaning from the text. The text is checked for meaningfulness. It is done by mapping syntactic structures and objects in the task domain. The semantic analyzer disregards sentence such as “hot ice-cream”.

•**Discourse Integration** – The meaning of any sentence depends upon the meaning of the sentence just before it. In addition, it also brings about the meaning of immediately succeeding sentence.

•**Pragmatic Analysis** – During this, what was said is re-interpreted on what it actually meant. It involves deriving those aspects of language which require real world knowledge.

Difficulties in NLU

NL has an extremely rich form and structure.

It is very ambiguous. There can be different levels of ambiguity –

•**Lexical ambiguity** – It is at very primitive level such as word-level.

•For example, treating the word “board” as noun or verb?

- **Syntax Level ambiguity** – A sentence can be parsed in different ways.
- For example, “He lifted the beetle with red cap.” – Did he use cap to lift the beetle or he lifted a beetle that had red cap?
- **Referential ambiguity** – Referring to something using pronouns. For example, Rima went to Gauri. She said, “I am tired.” – Exactly who is tired?
- One input can mean different meanings.
- Many inputs can mean the same thing.

Importance of NLP

The advantage of natural language processing can be seen when considering the following two statements: "Cloud computing insurance should be part of every service level agreement" and "A good SLA ensures an easier night's sleep -- even in the cloud." If you use natural language processing for search, the program will recognize that *cloud computing* is an entity, that *cloud* is an abbreviated form of cloud computing and that *SLA* is an industry acronym for service level agreement.

These are the types of vague elements that frequently appear in human language and that machine learning algorithms have historically been bad at interpreting. Now, with improvements in deep learning and artificial intelligence, algorithms can effectively interpret them.

This has implications for the types of data that can be analyzed. More and more information is being created online every day, and a lot of it is natural human language. Until recently, businesses have been unable to analyze this data. But advances in NLP make it possible to analyze and learn from a greater range of data sources.

Benefits of NLP

NLP hosts benefits such as:

- Improved accuracy and efficiency of documentation.
- The ability to automatically make a readable summary text.
- Useful for personal assistants such as Alexa.
- Allows an organization to use [chatbots](#) for customer support.
- Easier to perform [sentiment analysis](#).

Natural Language Processing (NLP) Applications

Business Applications for Natural Language Processing

Let's start with the domain of commercialization. It is pretty evident that the business domain consists of some interesting and necessary use cases and problems which can be addressed by the use of Natural Language Processing. Some use cases of Natural Language Processing which are used in the Business domain are –

Sentiment Analysis – It is widely used in social media analytics and web monitoring which allow knowing the insights of the customers concerning particular products or services. It can be advantageous for any company to know about the thinking of the customers about a product so that they can know about the scope of improvement and how to achieve robustness. Natural language processing can not solely handle this task; it requires integration with highly computational methods such as Machine learning and deep learning to do the back end computation and Big data analytics to digest the data at an enormous scale.

Email Filters – Emails are adopted as a medium of communication officially now. Even the government considers it official to communicate with the help of Email. But this medium is also vulnerable to spamming of the content. Companies which provide Email domains such as Google, Zoho or Yahoo are researching in the field of making it Full-proof by using different measures. Email filtering is an everyday use case of Natural Language Processing by applying various text analytics measures. It is a task of spam detection which is also in sentiment analysis as a pre-processing technique.

Voice Recognition – These are techniques which are powered by Natural Language Processing that allow companies to develop smart voice-driven services and interfaces for any product and service. To narrow the communication gap between the machines and human is the most critical and necessary step to increase the grip on Artificial Intelligence. It can be achieved by only and only Voice Recognition which is possible by Natural Language Understanding a sub-process of Natural Language Processing.

Information Extraction – Information is the new fuel, it is a well-known fact now. But the data which is received at any receiving end mostly consists of unstructured format. The emergence of the advanced statistical algorithms results in the rise predictive analytics and prescriptive analytics which made the prediction system more accurate. But these algorithms demand more and more information for finding the patterns and Matching them. Of course, Machine

learning and Deep learning methods are doing an impressively great job, but without Natural Language Processing these things are not possible.

Role of Natural Language Processing in Healthcare

Healthcare is the domain where accuracy and efficiency both are required because it is directly related to the health of the human kind, so the margin of error is approximately near to zero. Natural Language Processing addressed the issues of healthcare with some of its application and use cases.

Raising the bar of provider interactions with Patients and EHR

The main concern and priority in nowadays the healthcare system is to provide better and 24/7 EHR experience. It is referred to ensure unparalleled attention which also gets influenced by the demand of completing the documentation work, results in the dissatisfaction of the customer toward a clinic or hospital. This problem is on the way of becoming an epidemic for healthcare. But it can be prevented with the use of NLP which can be used mainly in three ways.

- Intelligent Voice Support Systems**
- Predictive Analytics**
- Prescriptive Analytics**

For Example – In Pennsylvania, Well Span Health started using voice-based tools for dictating the patient-provider interactions which reduced the frustration of EHR.

Raising the bar of patient health literacy

NLP can also be used to reduce the communication and interaction gap between Healthcare technologies (such as patient portals which contain health records of a patient) and patients. Health care domain already started to adopt the technology in various ways, but the patient still finds it hard to take that. NLP can be a tool for them. Suppose a system attached with a Healthcare portal with which a patient can interact with his/her native language. This will result in three advantages.

- It will become easy for every patient to understand his/her health status.**
- It reduces the chances of Human error in the system.**
- It will give a comfortable space to doctors too.**

Increasing the dimension of high quality of care.

This point can be considered as the extension of the second point mentioned above. Healthcare reports generally contain parameters which require proper

attention. Sometimes human error cause causality which can be eliminated using the machine (or computer devices, in simple words) which demand the use of Natural Language Processing. A study proved (done in 2018) the use of NLP can provide significant relief in the case of calculating the measure of inpatient care and monitoring the clinical guidelines.

Identification of the patients which require Improved Care Coordination

Identification of the patients here refers to the proper identification of the diseases present in any human. In this task NLP integrated with Machine learning have shown great potential. Automated Detection of Cancer, Detection of the root causes related to any substance disorder are some of the examples. This process can be done by extracting the information from old existed data set using NLP and using this information for training Machine Learning and Deep Learning models.

For Example – At Massachusetts General Hospital, the researcher used applied NLP techniques to identify the main reasons associated with the social determinants of health.

Natural Language Processing Applications in Finance

Credit Scoring Model/Method

Credit scoring is a risk estimation method (estimate risk while providing the loan to any party) in which risk of giving loan and credit is calculated by the help of the credit score against the credit histories of the potential borrowers. Natural Language Processing (integrated with AI) related to credit scoring more often than not are predictive analytics solutions. Natural Language Processing is also used by some companies to mine the social media of the customers. The data extracted from social media is then used to likely weighs certain online behaviors.

For Example, A Singapore-based company named as Lenddo EFL (with 115 employees) developed software called Lenddo Score which use machine learning and NLP to assess and calculate an individual's creditworthiness.

Document Search Engine

Natural Language Processing also increased the level of the Information extraction from structured and unstructured data which resulted in a big plus in the field of Documentation Processes.

For example, A private firm named as Nuance Communications based in Massachusetts developed software known as Nuance Document Finance

Solution, which is used to aid financial services companies in automatizing the documentation process.

Fraud Detection in Banking

The world of finance is very vulnerable to fraud and this world solely based on the text as every entry and record is maintained in the form of text. Natural Language Processing with the help of Machine Learning is used to detect fraud and misinterpreted information. Natural Language Processing generally used to extract information and process information which is further used by Machine learning model to train themselves to expose the fraud.

Defense and National Security

USA 's Defence Research and Analysis wing [DARPA](#) (Defence Advanced Research Projects Agency) developed a program DEFT (Deep Exploration and Filtering of Text) in which Natural Language Processing is used to extract pertinent information from unstructured data. This information is further used to analytics procedures to draw some insights from the data. Moreover, NLP based models also used by the Institute for Strategic Dialogue in the United Kingdom to observe and trace the signs of radicalization and extremism.

Natural Language Processing in Recruitment

Natural Language Processing can be used to perform the text analytics for searching the appropriate applications from the data, and it also can be used for selecting the best applications from the data available. Natural Language Processing can be used on different phases and with a different medium. Some of the primary use cases which can be used in Recruitment are Information Extraction, Social Media Analytics, Fraud Detection, and Voice Support systems.

2. Naive Bayes Algorithm:

It is a [classification technique](#) based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

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

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

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

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

Above,

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

How Naive Bayes algorithm works?

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

Step 1: Convert the data set into a frequency table

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

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
All	5	9
	$\approx 5/14$	$\approx 9/14$
	0.36	0.64

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

Problem: Players will play if weather is sunny. Is this statement is correct?

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

$$P(\text{Yes} \mid \text{Sunny}) = P(\text{Sunny} \mid \text{Yes}) * P(\text{Yes}) / P(\text{Sunny})$$

Here we have $P(\text{Sunny} \mid \text{Yes}) = 3/9 = 0.33$, $P(\text{Sunny}) = 5/14 = 0.36$, $P(\text{Yes}) = 9/14 = 0.64$

Now, $P(\text{Yes} \mid \text{Sunny}) = 0.33 * 0.64 / 0.36 = 0.60$, which has higher probability.

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

What are the Pros and Cons of Naive Bayes?

Pros:

- It is easy and fast to predict class of test data set. It also perform well in multi class prediction
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.

- It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

Cons:

- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.

- On the other side naive Bayes is also known as a bad estimator, so the probability outputs from predict_proba are not to be taken too seriously.

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

4 Applications of Naive Bayes Algorithms

- Real time Prediction:** Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.

- Multi class Prediction:** This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.

- Text classification/ Spam Filtering/ Sentiment Analysis:** Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)

- Recommendation System:** Naive Bayes Classifier and [Collaborative Filtering](#) together builds a Recommendation System that uses machine

learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not

3. When is unsupervised learning used over supervised learning?

Unsupervised learning is a machine learning technique, where you do not need to supervise the model. Instead, you need to allow the model to work on its own to discover information. It mainly deals with the unlabelled data.

Unsupervised learning algorithms allow you to perform more complex processing tasks compared to supervised learning. Although, unsupervised learning can be more unpredictable compared with other natural learning deep learning and reinforcement learning methods.

But In Supervised learning, you train the machine using data which is well "labeled." It means some data is already tagged with the correct answer. It can be compared to learning which takes place in the presence of a supervisor or a teacher.

A supervised learning algorithm learns from labeled training data, helps you to predict outcomes for unforeseen data. Successfully building, scaling, and deploying accurate supervised machine learning Data science model takes time and technical expertise from a team of highly skilled data scientists. Moreover, Data scientist must rebuild models to make sure the insights given remains true until its data changes.

Supervised machine learning is the more commonly used between the two. It includes such algorithms as linear and logistic regression, multi-class classification, and support vector machines. Supervised learning is so named because the data scientist acts as a guide to teach the algorithm what conclusions it should come up with. It's similar to the way a child might learn arithmetic from a teacher. Supervised learning requires that the algorithm's possible outputs are already known and 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.

On the other hand, unsupervised machine learning is more closely aligned with what some call true artificial intelligence — the idea that a computer can learn to identify complex processes and patterns without a human to provide guidance along the way. Although unsupervised learning is prohibitively complex for some simpler [enterprise use cases](#), it opens the doors to solving problems that humans normally would not tackle. Some examples of unsupervised machine learning algorithms include [k-means clustering](#), principal and independent component analysis, and association rules.

While a supervised classification algorithm learns to ascribe inputted labels to images of animals, its unsupervised counterpart will look at inherent similarities between the images and separate them into groups accordingly, assigning its own new label to each group. In a practical example, this type of algorithm is useful for customer segmentation because it will return groups based on parameters that a human may not consider due to pre-existing biases about the company's demographic.

Choosing to use either a supervised or unsupervised machine learning algorithm typically depends on factors related to the structure and volume of your data and the use case of the issue at hand. A well-rounded data science program will use both types of algorithms to build [predictive data models](#) that help stakeholders make decisions across a variety of business challenges.

Here, are prime reasons for using Unsupervised Learning:

- Unsupervised machine learning finds all kind of unknown patterns in data.
- Unsupervised methods help you to find features which can be useful for categorization.
- It is taken place in real time, so all the input data to be analyzed and labeled in the presence of learners.
- It is easier to get unlabeled data from a computer than labeled data, which needs manual intervention.

HOW supervised learning works?

For example, you want to train a machine to help you predict how long it will take you to drive home from your workplace. Here, you start by creating a set of labeled data. This data includes

- Weather conditions
- Time of the day
- Holidays

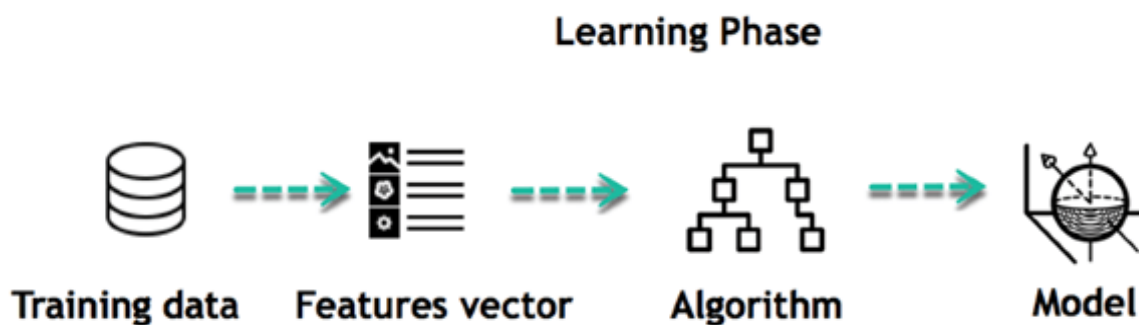
All these details are your inputs. The output is the amount of time it took to drive back home on that specific day.

You instinctively know that if it's raining outside, then it will take you longer to drive home. But the machine needs data and statistics.

Let's see now how you can develop a supervised learning model of this example which help the user to determine the commute time. The first thing you requires to create is a training data set. This training set will contain the total commute time and corresponding factors like weather, time, etc. Based on this training set, your machine might see there's a direct relationship between the amount of rain and time you will take to get home.

So, it ascertains that the more it rains, the longer you will be driving to get back to your home. It might also see the connection between the time you leave work and the time you'll be on the road.

The closer you're to 6 p.m. the longer time it takes for you to get home. Your machine may find some of the relationships with your labeled data.



This is the start of your Data Model. It begins to impact how rain impacts the way people drive. It also starts to see that more people travel during a particular time of day.

How Unsupervised Learning works?

Let's, take the case of a baby and her family dog.

She knows and identifies this dog. A few weeks later a family friend brings along a dog and tries to play with the baby.

Baby has not seen this dog earlier. But it recognizes many features (2 ears, eyes, walking on 4 legs) are like her pet dog. She identifies a new animal like a dog. This is unsupervised learning, where you are not taught but you learn from the data (in this case data about a dog.) Had this been supervised learning, the family friend would have told the baby that it's a dog.

Parameters	Supervised machine learning technique	Unsupervised machine learning technique
Process	In a supervised learning model, input and output variables will be given.	In unsupervised learning model, only input data will be given
Input Data	Algorithms are trained using labeled data.	Algorithms are used against data which is not labeled
Algorithms Used	Support vector machine, Neural network, Linear and logistics regression, random forest, and Classification trees.	Unsupervised algorithms can be divided into different categories: like Cluster algorithms, K-means, Hierarchical clustering, etc.
Computational Complexity	Supervised learning is a simpler method.	Unsupervised learning is computationally complex
Use of Data	Supervised learning model uses training data to learn a link between the input and the outputs.	Unsupervised learning does not use output data.
Accuracy of Results	Highly accurate and trustworthy method.	Less accurate and trustworthy method.
Real Time Learning	Learning method takes place offline.	Learning method takes place in real time.
Number of Classes	Number of classes is known.	Number of classes is not known.
Main Drawback	Classifying big data can be a real challenge in Supervised Learning.	You cannot get precise information regarding data sorting, and the output as data used in unsupervised learning is labeled and not known.

4. List various language-processing tasks, which can be done easily using NLTK.

- 1. Tokenization**
- 2. Sentence boundary detection**[\[1\]](#)
- 3. Shallow parsing**
- 4. Syntax parsing**
- 5. Semantics**
- 6. Pragmatics**
- 7. Named-entity**

- 8.Information extraction**
- 9.Terminology extraction**
- 10.Discourse parsing**
- 11.Topic modeling**
- 12.Summarizing**
- 13.Similarity**
- 14.Natural language generation**
- 15.Speech recognition**
- 16. Speech synthesis**
- 17. Ontology population**
- 18. Question answering**
- 19.Machine translation**
- 20.Search engine**
- 21.Fake news detection**

5. Explain step by step how would you classify sentiments of articles.

Explain data mining methods and machine learning algorithm in the process.

Step 1 — Installing NLTK and Downloading the Data

In this step you will install NLTK and download the sample tweets that you will use to train and test your model.

First, install the NLTK package with the pip package manager:

pip install nltk==3.3

Step 2 — Tokenizing the Data

Language in its original form cannot be accurately processed by a machine, so you need to process the language to make it easier for the machine to understand. The first part of making sense of the data is through a process called tokenization, or splitting strings into smaller parts called tokens.

Step 3 — Normalizing the Data

Words have different forms—for instance, “ran”, “runs”, and “running” are various forms of the same verb, “run”. Depending on the requirement of your analysis, all of these versions may need to be converted to the same form, “run”. **Normalization** in NLP is the process of converting a word to its canonical form.

Normalization helps group together words with the same meaning but different forms. Without normalization, “ran”, “runs”, and “running” would be treated as different words, even though you may want them to be treated as the same word. In this section, you explore stemming and lemmatization, which are two popular techniques of normalization.

Stemming is a process of removing affixes from a word. Stemming, working with only simple verb forms, is a heuristic process that removes the ends of words.

In this tutorial you will use the process of lemmatization, which normalizes a word with the context of vocabulary and **morphological analysis** of words in text. The lemmatization algorithm analyzes the structure of the word and its context to convert it to a normalized form. Therefore, it comes at a cost of speed.

A comparison of stemming and lemmatization ultimately comes down to a trade off between speed and accuracy.

Step 4 — Removing Noise from the Data

In this step, you will remove noise from the dataset. Noise is any part of the text that does not add meaning or information to data.

Noise is specific to each project, so what constitutes noise in one project may not be in a different project. For instance, the most common words in a language are called stop words. Some examples of stop words are “is”, “the”, and “a”. They are generally irrelevant when processing language, unless a specific use case warrants their inclusion.

Step 5 — Determining Word Density

The most basic form of analysis on textual data is to take out the word frequency. A single tweet is too small of an entity to find out the distribution of words, hence, the analysis of the frequency of words would be done on all positive tweets.

Step 6 — Preparing Data for the Model

Sentiment analysis is a process of identifying an attitude of the author on a topic that is being written about. You will create a training data set to train a model. It is a supervised learning machine learning process, which requires you to associate each dataset with a “sentiment” for training. In this tutorial, your model will use the “positive” and “negative” sentiments.

Sentiment analysis can be used to categorize text into a variety of sentiments. For simplicity and availability of the training dataset, this tutorial helps you train your model in only two categories, positive and negative.

A model is a description of a system using rules and equations. It may be as simple as an equation which predicts the weight of a person, given their height. A sentiment analysis model that you will build would associate tweets with a positive or a negative sentiment. You will need to split your dataset into two parts. The purpose of the first part is to build the model, whereas the next part tests the performance of the model.

Step 7 — Building and Testing the Model

Finally, you can use the `NaiveBayesClassifier` class to build the model. Use the `.train()` method to train the model and the `.accuracy()` method to test the model on the testing data.

```
from nltk import classify
```

```
from nltk import NaiveBayesClassifier  
classifier = NaiveBayesClassifier.train(train_data)
```

```
print("Accuracy is:", classify.accuracy(classifier, test_data))
```

```
print(classifier.show_most_informative_features(10))
```

Save, close, and execute the file after adding the code. The output of the code will be as follows:

Output

Accuracy is: 0.9956666666666667

Accuracy is defined as the percentage of tweets in the testing dataset for which the model was correctly able to predict the sentiment. A 99.5% accuracy on the test set is pretty good.

In the table that shows the most informative features, every row in the output shows the ratio of occurrence of a token in positive and negative tagged tweets in the training dataset. The first row in the data signifies that in all tweets containing the token :(, the ratio of negative to positives tweets was 2085.6 to 1. Interestingly, it seems that there was one token with :(in the positive datasets. You can see that the top two discriminating items in the text are the emoticons. Further, words such as sad lead to negative sentiments, whereas welcome and glad are associated with positive sentiments.

6. List at least 4 types of Machine learning techniques.

Here is the list of commonly used machine learning algorithms. These algorithms can be applied to almost any data problem:

- 1. Linear Regression**
- 2. Logistic Regression**
- 3. Decision Tree**
- 4. SVM**
- 5. Naive Bayes**
- 6. kNN**
- 7. K-Means**
- 8. Random Forest**
- 9. Dimensionality Reduction Algorithms**

1.linear Regression

It is used to estimate real values (cost of houses, number of calls, total sales etc.) based on continuous variable(s). Here, we establish relationship between independent and dependent variables by fitting a best line. This best fit line is known as regression line and represented by a linear equation $Y = a * X + b$.

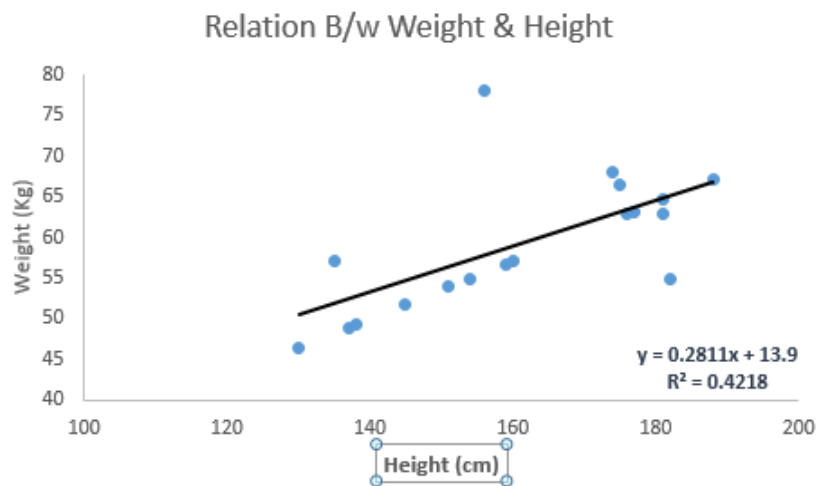
The best way to understand linear regression is to relive this experience of childhood. Let us say, you ask a child in fifth grade to arrange people in his class by increasing order of weight, without asking them their weights! What do you think the child will do? He / she would likely look (visually analyze) at the height and build of people and arrange them using a combination of these visible parameters. This is linear regression in real life! The child has actually figured out that height and build would be correlated to the weight by a relationship, which looks like the equation above.

In this equation:

- Y – Dependent Variable**
- a – Slope**
- X – Independent variable**
- b – Intercept**

These coefficients a and b are derived based on minimizing the sum of squared difference of distance between data points and regression line.

Look at the below example. Here we have identified the best fit line having linear equation $y = 0.2811x + 13.9$. Now using this equation, we can find the weight, knowing the height of a person.



Linear Regression is mainly of two types: Simple Linear Regression and Multiple Linear Regression. Simple Linear Regression is characterized by one independent variable. And, Multiple Linear Regression(as the name suggests) is characterized by multiple (more than 1) independent variables. While finding the best fit line, you can fit a polynomial or curvilinear regression. And these are known as polynomial or curvilinear regression.

2. logistic Regression:

Don't get confused by its name! It is a classification not a regression algorithm. It is used to estimate discrete values (Binary values like 0/1, yes/no, true/false) based on given set of independent variable(s). In simple words, it predicts the probability of occurrence of an event by fitting data to a [logit function](#). Hence, it is also known as logit regression. Since, it predicts the probability, its output values lies between 0 and 1 (as expected).

Again, let us try and understand this through a simple example.

Let's say your friend gives you a puzzle to solve. There are only 2 outcome scenarios – either you solve it or you don't. Now imagine, that you are being given wide range of puzzles / quizzes in an attempt to understand which subjects you are good at. The outcome to this study would be something like this – if you are given a trigonometry based tenth grade problem, you are 70% likely to solve it. On the other hand, if it is grade fifth history question, the probability of getting an answer is only 30%. This is what Logistic Regression provides you.

Coming to the math, the log odds of the outcome is modeled as a linear combination of the predictor variables.

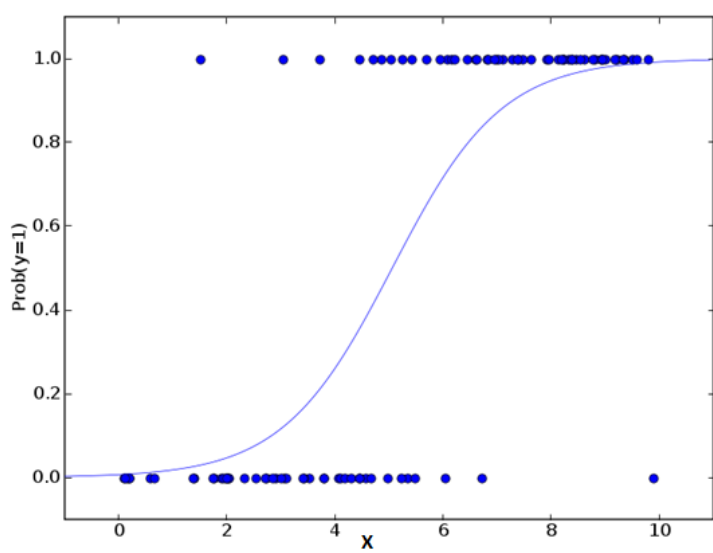
odds= $p / (1-p)$ = probability of event occurrence / probability of not event occurrence

$\ln(\text{odds}) = \ln(p/(1-p))$

$\text{logit}(p) = \ln(p/(1-p)) = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + \dots + b_kX_k$

Above, p is the probability of presence of the characteristic of interest. It chooses parameters that maximize the likelihood of observing the sample values rather than that minimize the sum of squared errors (like in ordinary regression).

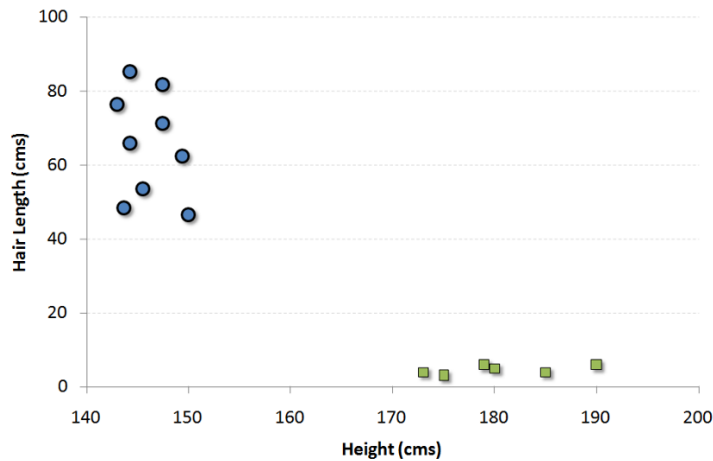
Now, you may ask, why take a log? For the sake of simplicity, let's just say that this is one of the best mathematical way to replicate a step function. I can go in more details, but that will beat the purpose of this article.



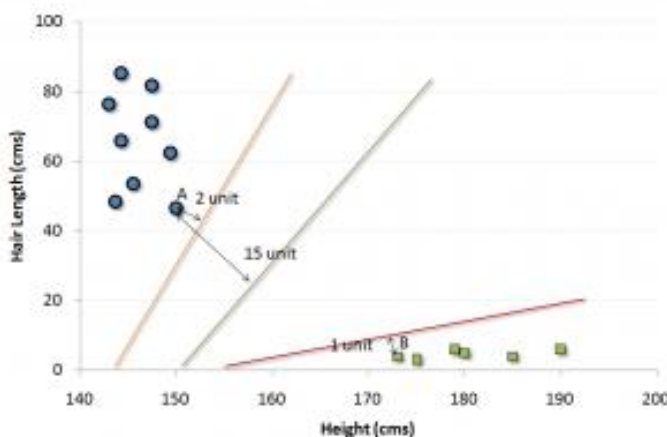
3.SVM:

It is a classification method. In this algorithm, we plot each data item as a point in n -dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate.

For example, if we only had two features like Height and Hair length of an individual, we'd first plot these two variables in two dimensional space where each point has two co-ordinates (these co-ordinates are known as Support Vectors)



Now, we will find some *line* that splits the data between the two differently classified groups of data. This will be the line such that the distances from the closest point in each of the two groups will be farthest away.



In the example shown above, the line which splits the data into two differently classified groups is the *black* line, since the two closest points are the farthest apart from the line. This line is our classifier. Then, depending on where the testing data lands on either side of the line, that's what class we can classify the new data as.

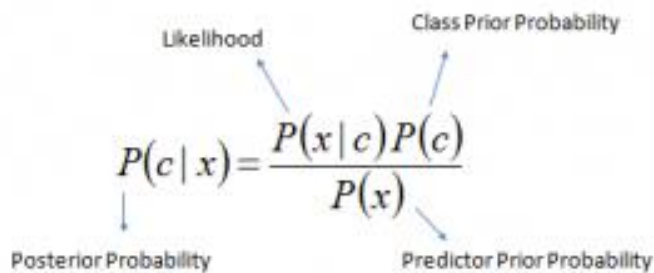
4. NAIVE BAYES”:

It is a classification technique based on [Bayes' theorem](#) with an assumption of independence between predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes

classifier would consider all of these properties to independently contribute to the probability that this fruit is an apple.

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

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



The diagram shows the equation $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$. Arrows point from labels to parts of the equation: 'Likelihood' points to $P(x|c)$, 'Class Prior Probability' points to $P(c)$, 'Posterior Probability' points to $P(c|x)$, and 'Predictor Prior Probability' points to $P(x)$.

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

Here,

- $P(c|x)$ is the posterior probability of *class (target)* given *predictor (attribute)*.
- $P(c)$ is the prior probability of *class*.
- $P(x/c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

Example: Let's understand it using an example. Below I have a training data set of weather and corresponding target variable 'Play'. Now, we need to classify whether players will play or not based on weather condition. Let's follow the below steps to perform it.

Step 1: Convert the data set to frequency table

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

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
All	5	9
	$\approx 5/14$	$\approx 9/14$
	0.36	0.64

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

Problem: Players will play if weather is sunny, is this statement is correct?

We can solve it using above discussed method, so $P(\text{Yes} | \text{Sunny}) = P(\text{Sunny} | \text{Yes}) * P(\text{Yes}) / P(\text{Sunny})$

Here we have $P(\text{Sunny} | \text{Yes}) = 3/9 = 0.33$, $P(\text{Sunny}) = 5/14 = 0.36$, $P(\text{Yes}) = 9/14 = 0.64$

Now, $P(\text{Yes} | \text{Sunny}) = 0.33 * 0.64 / 0.36 = 0.60$, which has higher probability.

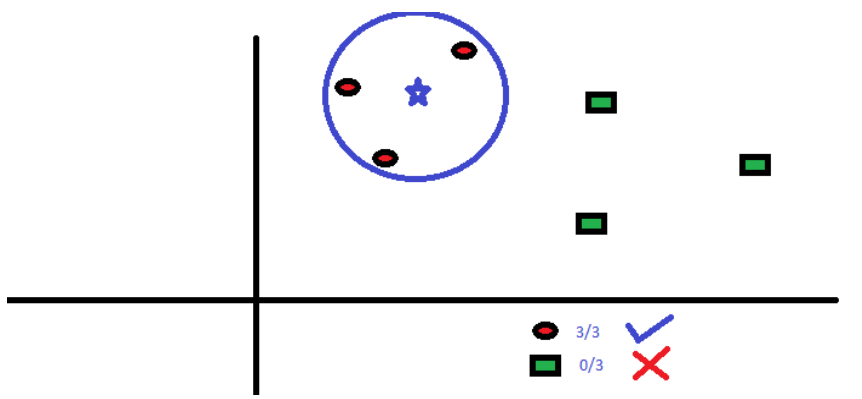
Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

5.KNN:

It can be used for both classification and regression problems. However, it is more widely used in classification problems in the industry. K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases by a majority vote of its k neighbors. The case being assigned to the class is most common amongst its K nearest neighbors measured by a distance function.

These distance functions can be Euclidean, Manhattan, Minkowski and Hamming distance. First three functions are used for continuous function and fourth one (Hamming) for categorical variables. If $K = 1$, then the case is simply assigned to the class of its nearest neighbor. At times, choosing K turns out to be a challenge while performing kNN modeling.

More: [Introduction to k-nearest neighbors : Simplified.](#)



KNN can easily be mapped to our real lives. If you want to learn about a person, of whom you have no information, you might like to find out about his close friends and the circles he moves in and gain access to his/her information!

Things to consider before selecting kNN:

- KNN is computationally expensive
- Variables should be normalized else higher range variables can bias it
- Works on pre-processing stage more before going for kNN like an outlier, noise removal

7. CORPUS:

Corpus is a large collection of texts. It is a body of written or spoken material upon which a linguistic analysis is based. The plural form of corpus is corpora. Some popular corpora are [British National Corpus](#) (BNC), COBUILD/Birmingham Corpus, IBM/Lancaster Spoken English Corpus. Monolingual corpora represent only one language while bilingual corpora represent two languages. European Corpus Initiative (ECI) corpus is multilingual having 98 million words in Turkish, Japanese, Russian, Chinese, and other languages. The corpus may be composed of written language, spoken language or both. Spoken corpus is usually in the form of audio recordings. A corpus may be open or closed. An *open corpus* is one which does not claim to contain all data from a specific area while a *closed corpus* does claim to contain all or nearly all data from a particular field. *Historical corpora*, for example, are closed as there can be no further input to an area.

USE OF CORPUS:

A corpus provides grammarians, lexicographers, and other interested parties with better descriptions of a language. Computer-processable corpora allow linguists to adopt the principle of total accountability, retrieving all the occurrences of a particular word or structure for inspection or randomly selected samples. Corpus analysis provides lexical information, morphosyntactic information, semantic information and pragmatic information.

APPLICATION OF CORPUS:

Corpora are used in the development of NLP tools. Applications include spell-checking, grammar-checking, speech recognition, text-to-speech and speech-to-text synthesis, automatic abstraction and indexing, information retrieval and machine translation. Corpora are also used for creation of new dictionaries and grammars for learners.

Lexicography

Corpus-derived frequency lists and, more especially, concordances are establishing themselves as basic tools for the **lexicographer**. . . .

Language Teaching

. . . The use of concordances as language-learning tools is currently a major interest in computer-assisted language learning (CALL; see Johns 1986). . . .

Speech Processing

Machine **translation** is one example of the application of corpora for what computer scientists call *natural language processing*. In addition to machine translation, a major research goal for NLP is *speech processing*, that is, the development of computer systems capable of outputting automatically produced speech from written input (*speech synthesis*), or converting speech input into written form (*speech recognition*)."

Advantages of Corpus Linguistics

- Corpus data are more objective than data based on introspection.
- Corpus data can easily be verified by other researchers and researchers can share the same data instead of always compiling their own.
- Corpus data are needed for studies of variation

between **dialects**, **registers** and **styles**.

- Corpus data provide the frequency of occurrence of linguistic items.
- Corpus data do not only provide illustrative examples, but are a theoretical resource.
- Corpus data give essential information for a number of applied areas, like language teaching and language technology (machine translation, speech synthesis etc.).
- Corpora provide the possibility of total accountability of linguistic features--the analyst should account for everything in the data, not just selected features.
- Computerised corpora give researchers all over the world access to the data.
- Corpus data are ideal for non-native speakers of the language

corpus is an important tool because Corpus analysis provides quantitative, reusable data, and an opportunity to test and challenge our ideas and intuitions about language. Further, analysis applied to corpora as transcriptions or other types of linguistic annotation can be checked for consistency and inter-annotator agreement, and the annotated corpus can be reviewed and reused by others. Corpora are essential in particular for the study of spoken and signed language: while written language can be studied by examining the text, speech, signs and gestures disappear when they have been produced and thus, we need multimodal corpora in order to study interactive face-toface communication.

8. DIFFERENT BETWEEN CORPUS AND DATABASE.

A corpus is a collection of texts, written or spoken, usually stored in a computer database. A corpus may be quite small, for example, containing only 50,000 words of text, or very large, containing many millions of words. ...

Written texts in corpora might be drawn from books, newspapers, or magazines that have been scanned or downloaded electronically. Other written corpora might contain works of literature, or all the writings of one author (e.g., William Shakespeare). Such corpora help us to see how language is used in contemporary society, how our use of language has changed over time, and how language is used in different situations.

Spoken corpora, on the other hand, contain transcripts of spoken language. Such transcripts may be of ordinary conversations recorded in people's homes and workplaces, or of phone calls, business meetings, radio broadcasts, or TV shows. Like written corpora, spoken corpora show us how language is used in real life and in many different contexts.

People build corpora of different sizes for specific reasons. For example, a very large corpus would be required to help in the preparation of a dictionary. It might contain tens of millions of words – because it has to include many examples of all the words and expressions that are used in the language. A medium-sized corpus might contain transcripts of lectures and seminars and could be used to write books for learners who need academic language for their studies. Such corpora range in size from a million words to five or ten million words. Other corpora are more specialized and much smaller. These might contain the transcripts of business meetings, for instance, and could be used to help writers design materials for teaching business language.

Once a corpus is stored in a database, we can analyze it and “search” for information in the same way we use search engines to find keywords on the Internet, but with more sophisticated tools. By searching a corpus we can get answers to questions like these:

- What are the most frequent words and phrases in English?
- What are the differences between spoken and written English?
- Which tenses do people use most frequently?
- What prepositions follow particular verbs?
- How do people use words like can, may and might?
- How often do people use idiomatic expressions and why?

With corpora and software tools to analyze them, we can see how language is really used. We no longer have to rely heavily on intuition to know what we say or what we write; instead we can see what hundreds of different speakers and writers have actually said or written, all at the click of a mouse.

A corpus, then, is simply a large collection of texts that we can analyze using computer software, just as we can access the millions of texts on the Internet. It is not a theory of language learning or a teaching methodology, but it does influence our way of thinking about language and the kinds of texts and examples we use in language teaching.

DATABASE:

A database is a collection of information that is organized so that it can be easily accessed, managed and updated. Computer databases typically contain

aggregations of data records or [files](#), containing information about sales transactions or interactions with specific customers.

In a [relational database](#), digital information about a specific customer is organized into rows, columns and tables which are indexed to make it easier to find relevant information through [SQL](#) or [NoSQL](#) queries. In contrast, a [graph database](#) uses nodes and edges to define relationships between data entries and queries require a special [semantic search](#) syntax. As of this writing, [SPARQL](#) is the only semantic query language that is approved by the World Wide Web Consortium ([W3C](#)).

Typically, the database manager provides users with the ability to control read/write access, specify report generation and analyze usage. Some databases offer [ACID](#) (atomicity, consistency, isolation and durability) compliance to guarantee that data is consistent and that transactions are complete.

Relational database

A [relational database](#), invented by [E.F. Codd](#) at IBM in 1970, is a tabular database in which data is defined so that it can be reorganized and accessed in a number of different ways.

Relational databases are made up of a set of tables with data that fits into a predefined category. Each table has at least one data category in a column, and each row has a certain data instance for the categories which are defined in the columns.

The Structured Query Language (SQL) is the standard user and application program interface for a relational database. Relational databases are easy to extend, and a new data category can be added after the original database creation without requiring that you modify all the existing applications.

Distributed database

A distributed database is a database in which portions of the database are stored in multiple physical locations, and in which processing is dispersed or replicated among different points in a network.

Distributed databases can be homogeneous or heterogeneous. All the physical locations in a homogeneous [distributed database system](#) have the same underlying hardware and run the same operating systems and database applications. The hardware, operating systems or database applications in a heterogeneous distributed database may be different at each of the locations.

NoSQL database

NoSQL databases are useful for large sets of distributed data.

NoSQL databases are effective for big data performance issues that relational databases aren't built to solve. They are most effective when an organization must analyze large chunks of unstructured data or data that's stored across multiple virtual servers in the cloud.

Object-oriented database

Items created using object-oriented programming languages are often stored in relational databases, but object-oriented databases are well-suited for those items.

An object-oriented database is organized around objects rather than actions, and data rather than logic. For example, a multimedia record in a relational database can be a definable data object, as opposed to an alphanumeric value.

As nouns the difference between **corpus** and **database**

is that **corpus** is body while **database** is (computing) a collection of (usually) organized information in a regular structure, usually but not necessarily in a machine-readable format accessible by a computer.

8. Define prior probability:

a prior probability distribution, often simply called the prior, of an uncertain quantity is the **probability distribution** that would express one's beliefs about this quantity before some evidence is taken into account. For example, the prior could be the probability distribution representing the relative proportions of voters who will vote for a particular politician in a future election. The unknown quantity may be a **parameter** of the model or a **latent variable** rather than an **observable variable**.

Bayes' theorem calculates the renormalized pointwise product of the prior and the **likelihood function**, to produce the *posterior probability distribution*, which is the conditional distribution of the uncertain quantity given the data.

Similarly, the prior probability of a **random event** or an uncertain proposition is the **unconditional probability** that is assigned before any relevant evidence is taken into account.

Priors can be created using a number of methods.[1](pp27–41) A prior can be determined from past information, such as previous experiments. A prior can be *elicited* from the purely subjective assessment of an experienced expert.

An *uninformative prior* can be created to reflect a balance among outcomes when no information is available. Priors can also be chosen according to some principle, such as symmetry or maximizing entropy given constraints; examples are the **Jeffreys prior** or Bernardo's reference prior. When a family of *conjugate priors* exists, choosing a prior from that family simplifies calculation of the posterior distribution.

Parameters of prior distributions are a kind of *hyperparameter*. For example, if one uses a **beta distribution** to model the distribution of the parameter p of a **Bernoulli distribution**, then:

- p is a parameter of the underlying system (Bernoulli distribution), and
- α and β are parameters of the prior distribution (beta distribution); hence *hyperparameters*.

Hyperparameters themselves may have **hyperprior** distributions expressing beliefs about their values. A Bayesian model with more than one level of prior like this is called a **hierarchical Bayes model**.

The prior probability of an event will be revised as new data or information becomes available, to produce a more accurate measure of a potential outcome. That revised probability becomes the **posterior probability** and is calculated using **Bayes' theorem**. In statistical terms, the posterior probability is the probability of event A occurring given that event B has occurred.

For example, three acres of land have the labels A, B, and C. One acre has reserves of oil below its surface, while the other two do not. The prior probability of oil being found on acre C is one third, or 0.333. But if a drilling test is conducted on acre B, and the results indicate that no oil is present at the location, then the posterior probability of oil being found on acres A and C become 0.5, as each acre has one out of two chances.

Baye's theorem is a very common and fundamental theorem used in **data mining** and **machine learning**.

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A) \times P(B|A)}{P(B)}$$

$P(A)$ is the prior probability of A occurring

$P(A|B)$ is the conditional probability of A given that B occurs. This is the posterior probability due to its variable dependency on B. This assumes that the A is not independent of B.

$P(B|A)$ is the conditional probability of B given that A occurs.

$P(B)$ is the probability of B occurring.

If we are interested in the probability of an event of which we have prior observations; we call this the prior probability. We'll deem this event A, and its probability $P(A)$. If there is a second event that affects $P(A)$, which we'll call event B, then we want to know what the probability of A is given B has occurred. In probabilistic notation, this is $P(A|B)$, and is known as posterior probability or revised probability. This is because it has occurred after the original event, hence the post in posterior. This is how [Baye's theorem uniquely allows us](#) to update our previous beliefs with new information.

9.How is reinforced learning different from supervised learning?

Supervised Learning is the concept of machine learning that means the process of learning a practice of developing a function by itself by learning from a number of similar examples. This is a process of learning a generalized concept from few examples provided those of similar ones.

Reinforcement Learning is also an area of machine learning based on the concept of behavioral psychology that works on interacting directly with an environment which plays a key component in the area of Artificial Intelligence.

Supervised Learning and Reinforcement Learning comes under the area of Machine Learning which was coined by an American computing professional Arthur Samuel Lee in 1959 who is expert in Computer Gaming and Artificial Intelligence.

Machine Learning is a part of Computer Science where the capability of a software system or application will be improved by itself using only data instead of being programmed by programmers or coders.

In Machine Learning the performance capability or efficiency of a system improves itself by repeatedly performing the tasks by using data. Machine Learning also relates to computing, statistics, predictive analytics, etc.

difference between Supervised Learning and Reinforcement Learning

- 1. Supervised Learning has two main tasks called Regression and Classification whereas Reinforcement Learning has different tasks such as**

exploitation or exploration, Markov's decision processes, Policy Learning, Deep Learning and value learning.

- 2. Supervised Learning analyses the training data and produces a generalized formula, In Reinforcement Learning basic reinforcement is defined in the model Markov's Decision process.**
- 3. In Supervised Learning, each example will have a pair of input objects and an output with desired values whereas in Reinforcement Learning Markov's Decision process means the agent interacts with the environment in discrete steps i.e., agent makes an observation for every time period "t" and receives a reward for every observation and finally the goal is to collect as many rewards as possible to make more observations.**
- 4. In Supervised Learning, different numbers of algorithms exist with advantages and disadvantages that suit the system requirement. In Reinforcement Learning, Markov's decision process provides a mathematical framework for modeling and decision making situations.**
- 5. The most used learning algorithms for both Supervised learning and Reinforcement learning are linear regression, logistic regression, decision trees, Bayes Algorithm, Support Vector Machines, and Decision trees, etc., those which can be applied in different scenarios.**
- 6. In Supervised Learning, the goal is to learn the general formula from the given examples by analyzing the given inputs and outputs of a function. In Reinforcement Learning, the goal is in such way like controlling mechanism like control theory, gaming theory, etc., for example, driving a vehicle or playing gaming against another player, etc.,**
- 7. In Supervised learning both input and output will be available for decision making where the learner will be trained on many examples or sample data given whereas in reinforcement learning sequential decision making happens and the next input depends on the decision of the learner or system, examples are like playing chess against an opponent, robotic movement in an environment, gaming theory.**
- 8. In Supervised learning, just a generalized model is needed to classify data whereas in reinforcement learning the learner interacts with the environment to extract the output or make decisions, where the single output will be available in the initial state and output, will be of many possible solutions.**
- 9. Supervised learning means the name itself says it is highly supervised whereas the reinforcement learning is less supervised and depends on the learning agent in determining the output solutions by arriving at different possible ways in order to achieve the best possible solution.**

10. Supervised learning makes prediction depending on a class type whereas reinforcement learning is trained as a learning agent where it works as a reward and action system.
11. In Supervised learning, a huge amount of data is required to train the system for arriving at a generalized formula whereas in reinforcement learning the system or learning agent itself creates data on its own to by interacting with the environment.
12. Both Supervised learning and reinforcement learning are used to create and bring some innovations like robots that reflect human behavior and works like a human and interacting more with the environment causes more growth and development to the systems performance results in more technological advancement and growth.

Supervised Learning vs Reinforcement Learning Comparison Table

BASIS FOR

COMPARISON	Supervised Learning	Reinforcement learning
Definition	Works on existing or given sample data or examples	Works on interacting with the environment
Preference	Preferred in generalized working mechanisms where routine tasks are required to be done	Preferred in the area of Artificial Intelligence
Area	Comes under the area of Machine Learning	Comes under the area of Machine Learning
Platform	Operated with interactive software systems or applications	Supports and works better in Artificial Intelligence where Human Interaction is prevalent
Generality	Many open source projects are evolving of development in this area	More useful in Artificial Intelligence
Algorithm	Many algorithms exist in using this learning	Neither supervised nor unsupervised algorithms are used
Integration	Runs on any platform or with any applications	Runs with any hardware or software devices

REINFORCEMENT LEARNING:

Reinforcement learning is the training of machine learning models to [make a sequence of decisions](#). The agent learns to achieve a goal in an uncertain, potentially complex environment. In reinforcement learning, an artificial intelligence faces a game-like situation. The computer employs trial and error to come up with a solution to the problem. To get the machine to do what the programmer wants, the artificial intelligence gets either rewards or penalties for the actions it performs. Its goal is to maximize the total reward.

Although the designer sets the reward policy—that is, the rules of the game—he gives the model no hints or suggestions for how to solve the game. It's up to the model to figure out how to perform the task to maximize the reward, starting from totally random trials and finishing with sophisticated tactics and superhuman skills. By leveraging the power of search and many trials, reinforcement learning is currently the most effective way to hint machine's creativity. In contrast to human beings, artificial intelligence can gather experience from thousands of parallel gameplays if a reinforcement learning algorithm is run on a sufficiently powerful computer infrastructure.

Examples of reinforcement learning

Applications of reinforcement learning were in the past limited by weak computer infrastructure. However, as [Gerard Tesauro's backgamon AI superplayer developed in 1990's](#) shows, progress did happen. That early progress is now rapidly changing with powerful new computational technologies opening the way to completely new inspiring applications.

Training the models that control autonomous cars is an excellent example of a potential application of reinforcement learning. In an ideal situation, the computer should get no instructions on driving the car. The programmer would avoid hard-wiring anything connected with the task and allow the machine to learn from its own errors. In a perfect situation, the only hard-wired element would be the reward function.

- **For example**, in usual circumstances we would require an autonomous vehicle to put safety first, minimize ride time, reduce pollution, offer passengers comfort and obey the rules of law. With an autonomous race car, on the other hand, we would emphasize speed much more than the driver's comfort. The programmer cannot predict everything that could happen on the road. Instead of building lengthy "if-then" instructions, the programmer prepares the reinforcement learning agent to be capable of learning from the system of rewards and penalties. The agent (another name for reinforcement learning algorithms performing the task) gets rewards for reaching specific goals.

•**Another example:** [deepsense.ai](#) took part in the [“Learning to run” project](#), which aimed to train a virtual runner from scratch. The runner is an advanced and precise musculoskeletal model designed by the [Stanford Neuromuscular Biomechanics Laboratory](#). Learning the agent how to run is a first step in building a new generation of prosthetic legs, ones that automatically recognize people’s walking patterns and tweak themselves to make moving easier and more effective. While it is possible and has been [done in Stanford’s labs](#), hard-wiring all the commands and predicting all possible patterns of walking requires a lot of work from highly skilled programmers.

10.List Unix commands with its usage used in data manipulation.

awk	
Pattern scanning and processing language	
2	cmp Compares the contents of two files
3	comm Compares sorted data
4	cut Cuts out selected fields of each line of a file
5	diff

	Differential file comparator
6	expand Expands tabs to spaces
7	join Joins files on some common field
8	perl Data manipulation language
9	sed Stream text editor
10	sort Sorts file data
11	split Splits file into smaller files
12	tr Translates characters
13	uniq

	Reports repeated lines in a file
14	wc Counts words, lines, and characters
15	vi Opens vi text editor
16	vim Opens vim text editor
17	fmt Simple text formatter
18	spell Checks text for spelling error
19	ispell Checks text for spelling error
20	emacs GNU project Emacs
21	ex, edit

	Line editor
22	emacs GNU project Emacs

14. Why is POS-tagging important in language processing?

The part of speech explains how a word is used in a sentence. There are eight main parts of speech - nouns, pronouns, adjectives, verbs, adverbs, prepositions, conjunctions and interjections.

- **Noun (N)**- Daniel, London, table, dog, teacher, pen, city, happiness, hope
- **Verb (V)**- go, speak, run, eat, play, live, walk, have, like, are, is
- **Adjective(ADJ)**- big, happy, green, young, fun, crazy, three
- **Adverb(ADV)**- slowly, quietly, very, always, never, too, well, tomorrow
- **Preposition (P)**- at, on, in, from, with, near, between, about, under
- **Conjunction (CON)**- and, or, but, because, so, yet, unless, since, if
- **Pronoun(PRO)**- I, you, we, they, he, she, it, me, us, them, him, her, this
- **Interjection (INT)**- Ouch! Wow! Great! Help! Oh! Hey! Hi!

Most POS are divided into sub-classes. POS Tagging simply means labeling words with their appropriate Part-Of-Speech.

How does POS Tagging works?

[nlpforhackers](#)

POS tagging is a supervised learning solution that uses features like the previous word, next word, is first letter capitalized etc. NLTK has a function to get pos tags and it works after tokenization process.

POS tags make it possible for automatic text processing tools to take into account which part of speech each word is. This facilitates the use of linguistic criteria in addition to statistics.

For languages where the same word can have different parts of speech, e.g. *work* in English, POS tags are used to distinguish between the occurrences of the word when used as a noun or verb.

POS tags are also used to search for examples of grammatical or lexical patterns without specifying a concrete word, e.g. to find examples of any plural noun not preceded by an article.

Or both of the above can be combined, e.g. find the word *help* used as a noun followed by any verb in the past tense.

The Different POS Tagging Techniques

There are different techniques for POS Tagging:

- 1. Lexical Based Methods — Assigns the POS tag the most frequently occurring with a word in the training corpus.**
- 2. Rule-Based Methods — Assigns POS tags based on rules. For example, we can have a rule that says, words ending with “ed” or “ing” must be assigned to a verb. Rule-Based Techniques can be used along with Lexical Based approaches to allow POS Tagging of words that are not present in the training corpus but are there in the testing data.**
- 3. Probabilistic Methods — This method assigns the POS tags based on the probability of a particular tag sequence occurring. Conditional Random Fields (CRFs) and Hidden Markov Models (HMMs) are probabilistic approaches to assign a POS Tag.**
- 4. Deep Learning Methods — Recurrent Neural Networks can also be used for POS tagging.**

12. Why is there a need to “learn” for a machine?

Machine learning is the construction of algorithms and their application in the right scope. It is based on the idea of development of computer programs that can access data, analyse it and learn from it.

we cannot do everything, and machines can't not do what they're ordered.

They save both money and time by reducing human working hours.rowing volumes and varieties of data has made it difficult for coders to manually code and rectify every program. Here comes Machine Learning to ease human effort. It recognises and rectifies mistakes without human intervention, and also adapts with experience so that errors or bugs are never repeated in the future. This makes the whole process of coding and compiling faster, smoother and bug free. Moreover coders aren't available all the time, machines can update and upgrade algorithms every minute.

Extensive use of AI and Machine Learning proves it's importance. From cyber security to virtual assistants all have learning capabilities to serve human better. Recent years have shown that Machine Learning can be used to automate a lot of different tasks that were thought of as tasks that only humans can to like Image Recognition, Text Generation or playing games.

In 2014 Machine Learning and AI experts thought it would take at least 10 years before a machine could beat the world's best player at the board game Go. But Googles DeepMind proved them wrong. They showed that even in such a complex game as Go machines could learn which moves to consider. There are a lot more of advances in the field of machines playing games like the Dota Bot from the OpenAI Team.

Machine Learning is going to have huge effects on the economy and living in general. Entire work tasks and industries can be automated and the job market will be changed forever.

If you want to start learning about Machine Learning it's the perfect time because Machine Learning Engineers are desperately needed because a lot of companies want to get their foot in the door of Machine Learning and Artificial Intelligence.

Now that Machine Learning has all the attention it needs it's on us the Engineers and Researchers to drive for new big advances in the field of Machine Learning.

13.Is data mining an integral part of language processing? Explain

NLP is a way for computers to analyze, understand, and derive meaning from human language in a smart and useful way. By utilizing NLP, developers can organize and structure knowledge to perform tasks such as automatic summarization, translation, named entity recognition, relationship extraction, [sentiment analysis](#), speech recognition, and topic segmentation.

“Apart from common word processor operations that treat text like a mere sequence of symbols, NLP considers the hierarchical structure of language: several words make a phrase, several phrases make a sentence and, ultimately, sentences convey ideas,” John Rehling, an NLP expert at Meltwater Group, said in *[How Natural Language Processing Helps Uncover Social Media Sentiment](#)*. “By analyzing language for its meaning, NLP systems have long filled useful roles, such as correcting grammar, converting speech to text and automatically translating between languages.”

[NLP is used to analyze text](#), allowing machines to [understand how human's speak](#). This human-computer interaction enables real-world applications like [automatic text summarization](#), [sentiment analysis](#), [topic extraction](#), [named entity recognition](#), [parts-of-speech tagging](#), [relationship extraction](#), [stemming](#), and more. NLP is commonly used for [text mining](#), [machine translation](#), and [automated question answering](#).

NLP is characterized as a difficult problem in computer science. Human language is rarely precise, or plainly spoken. To understand human language is to understand not only the words, but the concepts and how they're [linked together to create meaning](#). Despite language being one of the easiest things for the human mind to learn, the ambiguity of language is what makes natural language processing a difficult problem for computers to master.

data mining is defined as a process used to extract usable data from a larger set of any raw data. It implies analysing data patterns in large batches of data using one or more software. Data mining has applications in multiple fields, like science and research. As an application of data mining, businesses can learn more about their customers and develop more effective strategies related to various business functions and in turn leverage resources in a more optimal and insightful manner. This helps businesses be closer to their objective and make better decisions. Data mining involves effective data collection and warehousing as well as computer processing. For segmenting the data and evaluating the

probability of future events, data mining uses sophisticated mathematical algorithms. Data mining is also known as Knowledge Discovery in Data (KDD).

The benefits of data mining come from the ability to uncover hidden patterns and relationships in data that can be used to make predictions that impact businesses.

Specific data mining benefits vary depending on the goal and the industry. Sales and marketing departments can mine customer data to improve lead conversion rates or to create [one-to-one marketing](#) campaigns. Data mining information on historical sales patterns and customer behaviors can be used to build prediction models for future sales, new products and services.

Companies in the financial industry use data mining tools to build risk models and detect fraud. The manufacturing industry uses data mining tools to improve product safety, identify quality issues, manage the [supply chain](#) and improve operations.

15. What does GREP and SED command do?

Grep (Global Regular Expression Processor) is a command used for searching for text, patterns (regex) in a file or a set of files. Let see the usage of Grep command using the following example.

country.txt

Afghanistan

Albania

Algeria

United States

Austria

Australia

Chile

Mongolia

China

France

Italy

India

Finland

Dubai

Indonesia

Armenia

To find the list of countries starting with 'a' use the following command

```
$ cat country.txt | grep 'A.*'
```

Afghanistan

Albania

Algeria

Austria

Australia

To do the above operation with case insensitive use the following command

```
$ cat country.txt | grep -i 'A.*'
```

Afghanistan

Albania

Algeria

Austria

Australia

Armenia

To find the number of lines in the file country.txt use the following command

```
$ cat country.txt | grep -c '.*'
```

17

To search for a particular country use the following command

```
$ cat country.txt | grep 'Austria'
```

Austria

You can also use the grep command without using the cat command as follows

```
$ grep 'Austria' country.txt
```

Austria

To find a match in all the files in a directory use the recursive grep command

```
$ grep -r "king" /home/examples/
```

To get the line number of a match in the file use the following command

```
$ grep -n 'Austria' country.txt
```

5:Austria

We can also use '-v' option to find the matches in only those lines which do not contain the given word (match).

```
$ cat country.txt | grep -v 'Austria'
```

Afghanistan

Albania

Algeria

United States

Australia

Chile

Mongolia

China

France

Italy

India

Finland

Dubai

Indonesia

Armenia

What is Sed?

SED (Stream Editor) is a powerful tool or command used for efficient text processing in Unix. Lets see the usage of SED command with the following examples.

To delete the line containing the given match use the following SED command

```
$ cat country.txt | sed -e '/Austria/d'
```

Afghanistan

Albania

Algeria

United States

Australia

Chile

Mongolia

China

France

Italy

India

Finland

Dubai

Indonesia

Armenia

The format to use SED command is

```
sed -<option> '/<match expression>/<command to be executed>'
```

The SED command gets the stream of input from stdin and then it fills the same in its pattern space.

Then the command after the slash '/Austria/d' (i.e) the 'd' which means delete

the line containing the given match 'Austria'. Once the command is executed the content of the pattern space is sent to the stdout.

Note that at the line in the actual file will not be deleted. You will be seeing the output only from the SED commands internal pattern space. To delete the line in the actual file then you need to redirect the output to the actual file as follows

```
$ cat country.txt | sed -e '/Austria/d' > countryNew.txt
```

To delete the first line in a file use the following command

```
$ cat country.txt | sed -e '1d'
```

Here you need not specify any match because we are going to delete only first line.

Albania

Algeria

United States

Austria

Australia

Chile

Mongolia

China

France

Italy

India

Finland

Dubai

Indonesia

Armenia

You can also specify the line number preceding the 'd' to delete that particular line.

Use the following command to delete blank lines in a file

```
$ cat country.txt | sed -e '/^$/d'
```

Search and Replace

To find a match and replace it with a given word, we can use the SED command in the following way

The format is as follows

Format:

```
sed -e 's/<find expression>/<replace expression>/' filename
```

Example:

```
$ sed -e 's/Austria/Australia/' country.txt
```

Output:

Afghanistan

Albania

Algeria

United States

Australia

Australia

Chile

Mongolia

China

France

Italy

India

Finland

Dubai

Indonesia

Armenia

To use the match as a part of replace string, we can use the following command

```
$ sed -n -e 's/United States/& of America/p' country.txt
```

United States of America

As you see the above output, ‘&’ has been used to represent the matched string, which is used as a part of replace string. So when the match ‘United States’ is found it is replace with ‘ United States of America’. ‘-n’ option is used so that the

entire output in the file is not printed. 'p' command used at the end of the command is used to print only the replaced match.

Convert text files to html files

Here let's do a simple conversion of text to html format. First create a file containing SED commands.

```
$ cat country.txt | sed -e '1i <html> <body>\
```

```
$ a </body> </html> ' > countryHtml.html
```

The command adds the HTML tags to the first line as 'i' is the insert command. Then '\$ a' moves to the end of the file content and adds the ending tags to the file content.

So the resulting file content will be as follows.

```
vi countryHtml.html
```

```
<html> <body>
```

```
Afghanistan
```

```
Albania
```

```
Algeria
```

```
United States
```

```
Austria
```

```
Australia
```

```
Chile
```

```
Mongolia
```

```
China
```

```
France
```

```
Italy
```

```
India
```

```
Finland
```

```
Dubai
```

```
Indonesia
```

```
Armenia
```

```
</body> </html>
```

In this way we can use SED command for text processing.

14.How would you classify a text without tagging POS in it?

Domain Specific Features in the Corpus

For a classification problem, it is important to choose the test and training corpus very carefully. For a variety of features to act in the classification algorithm, domain knowledge plays an integral part.

For example, if the problem is “Sentiment Classification on social media data”, the training corpus should consist of the data from social sources like twitter and facebook.

On the other hand if the problem is “Sentiment Classification for news data”, the corpus should consist of data from news sources. This is because the vocabulary of a corpus varies with domains. Social Media contains a lot of slangs and improper keywords like “awsum, lol, goood” etc which are absent in any of the formal corpus such as news, blogs etc.

Let’s take an example of a naive bayes classification problem where the task is to classify the statements into two Classes: “Class A and Class B”. We’ve training data corpus and a test data corpus. As mentioned here, the training corpus should contain the features from relevant corpus. Therefore, training corpus should consist of data points such as:

```
training_data = [  
    ...  
    ...  
    ('The apple iphone was gooooooooood, ##$! http://www.apple.com', 'Class  
A'),  
    ('I do not enjoy my job, holy shiiiit', 'Class B'),  
    ('I ain't feeling lol today crappp.', 'Class B'),  
    ('I feel amazing!'", 'Class A')  
    ...  
    ...  
]  
  
test_data = [  
    ('I luv this phones.', 'Class A'),
```

```
('This is an amaaaazingg company!', 'Class A'),  
( 'I am feeling very goood about these features lol.', 'Class A'),  
( 'This is my bestest phones.', 'Class A'),  
( "What an awesomee player", 'Class A'),  
( 'I do not like this phone #apple ', 'Class B'),  
( 'I am tired of this stuff.', 'Class B'),  
( 'They are my worst fears! . check them out here: http://goo.gl/qdjk3rf ',  
'Class B'),  
( 'My boss lives in India, He is horrible.', 'Class B')  
]
```

Stopwords are defined as the most commonly used words in a corpus. Most commonly used stopwords are “a, the, of, on, ... etc”. These words are used to define the structure of a sentence. But, are of no use in defining the context. Treating these type of words as feature words would result in poor performance in text classification. These words can be directly ignored from the corpus in order to obtain a better performance. Apart from language stopwords, There are some other supporting words as well which are of lesser importance than any other terms. These includes:

Language Stopwords – a, of, on, the ... etc

Location Stopwords – Country names, Cities names etc

Time Stopwords – Name of the months and days (january, february, monday, tuesday, today, tomorrow ...) etc

Numerals Stopwords – Words describing numerical terms (hundred, thousand, ... etc)

After removal of these entities from the test data, test data would be reformed to following:

```
test_data = [  
    ('luv phones.', 'Class A'),  
    (' amaaaazingg company!', 'Class A'),  
    (' feeling very goood about features lol.', 'Class A'),  
    (' bestest phones.', 'Class A'),  
    (" awesomee player", 'Class A'),  
    (' not like phone #apple ', 'Class B'),  
    (' tired stuff.', 'Class B'),  
]
```

(' worst fears! . check out here: <http://goo.gl/qdjk3rf> ', 'Class B'),
(' boss lives,horrible.', 'Class B')
x]

15. Why is it called 'natural language' and not just 'language'?

a natural language or ordinary language is any **language** that has **evolved** naturally in **humans** through use and repetition without conscious planning or premeditation. Natural languages can take different forms, such as **speech** or **signing**. They are distinguished from **constructed** and **formal languages** such as **those used to program computers** or to study **logic**. 'natural language' is used in opposition to the terms 'formal language' and 'artificial language,' but the important difference is that natural languages are not *actually constructed* as artificial languages and they do not *actually appear* as formal languages. But they are considered and studied as though they were formal languages 'in principle.' Behind the complex and the seemingly chaotic surface of natural languages there are--according to this way of thinking--rules and principles that determine their constitution and functions
natural language is unbounded is one of its more widely remarked upon properties and a core tenet of modern **linguistic theory**. The classic argument for creativity uses the idea that one can continually add further adjuncts to sentences to establish that there can be no longest **sentence** and therefore no finite number of sentences