Natural Language Processing

Introduction

Till now, we have explored two domains of AI: Data Science and Computer Vision. Both these domains differ from each other in terms of the data on which they work. Data Science works around numbers and tabular data while Computer Vision is all about visual data like images and videos. The third domain, Natural Language Processing (commonly called NLP) takes in the data of Natural Languages which humans use in their daily lives and operates on this.

Natural Language Processing, or NLP, is the sub-field of AI that is focused on enabling computers to understand and process human languages. AI is a subfield of Linguistics, Computer Science, Information Engineering, and Artificial Intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to process and analyse large amounts of natural language data.

But how do computers do that? How do they understand what we say in our language? This chapter is all about demystifying the Natural Language Processing domain and understanding how it works.

Before we get deeper into NLP, let us experience it with the help of this AI Game:

Identify the mystery animal: http://bit.ly/iai4yma

Go to this link on Google Chrome, launch the experiment and try to identify the Mystery Animal by asking the machine 20 Yes or No questions.

Were you able to guess the animal?

_____

_____

If yes, in how many questions were you able to guess it?

_____

_____

If no, how many times did you try playing this game?

_____

_____

What according to you was the task of the machine?

_____

_____

_____

_____

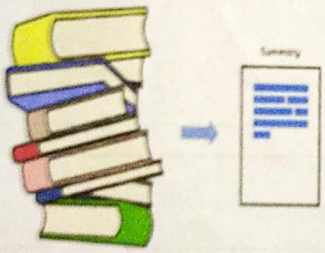
* Images shown here are the property of individual organisations and are used here for reference purpose only.

Were there any challenges that you faced while playing this game? If yes, list them down.

What approach must one follow to win this game?

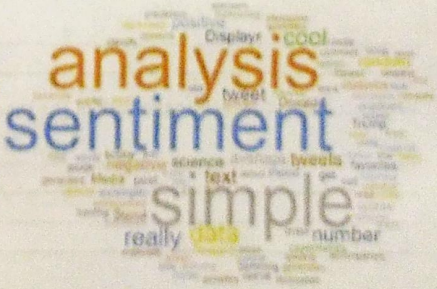
Applications of Natural Language Processing

Since Artificial Intelligence nowadays is becoming an integral part of our lives, its applications are very commonly used by the majority of people in their daily lives. Here are some of the applications of Natural Language Processing which are used in the real-life scenario:



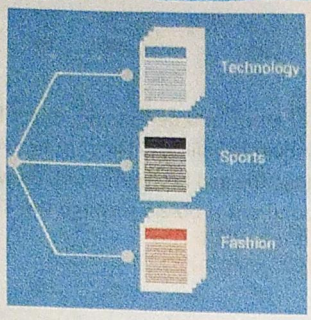
Automatic Summarization: Information overload is a real problem when we need to access a specific, important piece of information from a huge knowledge base. Automatic summarization is relevant not only for summarizing the meaning of documents and information, but also to understand the emotional meanings within the information, such as in collecting data from social media. Automatic summarization is especially relevant when used to provide an overview of a news item or blog post, while avoiding redundancy from multiple sources and maximizing the diversity of content obtained.

Sentiment Analysis: The goal of sentiment analysis is to identify sentiment among several posts or even in the same post where emotion is not always explicitly expressed. Companies use Natural Language Processing applications, such as sentiment analysis, to identify opinions and sentiment online to help them understand what customers think about their products and services (i.e., "I love the new iPhone" and, a few lines later "But sometimes it doesn't work well" where the person is still talking about the iPhone) and overall



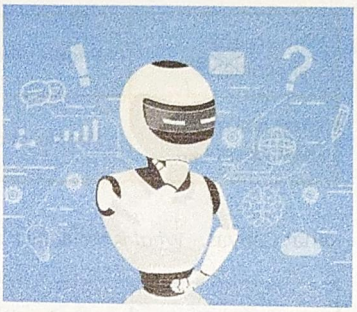
* Images shown here are the property of individual organisations and are used here for reference purpose only.

indicators of their reputation. Beyond determining simple polarity, sentiment analysis understands sentiment in context to help better understand what's behind an expressed opinion, which can be extremely relevant in understanding and driving purchasing decisions.



Text classification: Text classification makes it possible to assign predefined categories to a document and organize it to help you find the information you need or simplify some activities. For example, an application of text categorization is spam filtering in email.

Virtual Assistants: Nowadays Google Assistant, Cortana, Siri, Alexa, etc have become an integral part of our lives. Not only can we talk to them but they also have the abilities to make our lives easier. By accessing our data, they can help us in keeping notes of our tasks, make calls for us, send messages and a lot more. With the help of speech recognition, these assistants can not only detect our speech but can also make sense out of it. According to recent researches, a lot more advancements are expected in this field in the near future.



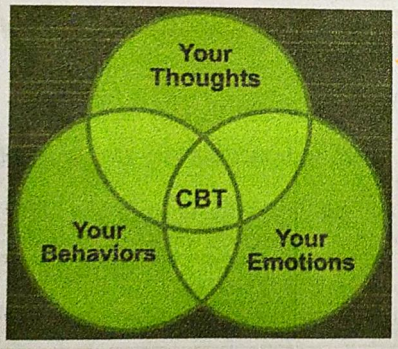
Natural Language Processing: Getting Started

Natural Language Processing is all about how machines try to understand and interpret human language and operate accordingly. But how can Natural Language Processing be used to solve the problems around us? Let us take a look.

Revisiting the AI Project Cycle

Let us try to understand how we can develop a project in Natural Language processing with the help of an example.

The Scenario



The world is competitive nowadays. People face competition in even the tiniest tasks and are expected to give their best at every point in time. When people are unable to meet these expectations, they get stressed and could even go into depression. We get to hear a lot of cases where people are depressed due to reasons like peer pressure, studies, family issues, relationships, etc. and they eventually get into something that is bad for them as well as for others. So, to overcome this, cognitive behavioural therapy (CBT) is considered to be one of the best methods to address stress as it is easy to implement on people and also gives good results. This therapy includes

* Images shown here are the property of individual organisations and are used here for reference purpose only.

Date : ____/____/____

Applications of NLP

1. Automatic Summarization :- We often face information overload when trying to access a specific piece of information from a huge Knowledge base. Automatic summarization summarises information within documents and data and understands emotional meaning within said documents and data.

Use in News Sites :- This technology helps provide a overview of a news item or blog post while avoiding redundancy from multiple sources and maximizing diversity of the content.

2. Sentiment Analysis :- Sentiment Analysis identifies sentiment among several posts or even in the same post where emotion is not always explicitly expressed.

Use by Companies :- Companies use NLP applications such as sentiment analysis to identify opinions and sentiments online to help them understand consumer perception of their products services and their overall Reputation.

For ex :- "I love the new app" but few lines later it says, "the update is buggy and doesn't always work well" where the person is still talking About the app.

Sentiment Analysis can determine polarity of opinion but beyond determing the simple polarity it can determine the sentiment behind a expressed opinion This is extremely relevant in understanding and driving purchase decisions.

Date : ____/____/____

3. Text Classification :- Text Classification is a NLP Application that assigns predefined categories to a document to help organise it to help you find the relevant information you need, or simplify some activities.

For eg:- Spam Filtering in Email.

4. Virtual Assistants :- Nowadays Virtual Assistants are popular and readily available such as Google Assistant, Siri, Alexa, Cortana and More. Not only can we talk to them, but they also make our lives easier. By accessing our data they help in keeping notes of our tasks, make calls for us, send messages etc.

* Using Speech Recognition, they can not only detect our speech, but also make sense out of it.

Therapy Includes

understanding the behaviour and mindset of a person in their normal life. With the help of CBT, therapists help people overcome their stress and live a happy life.
To understand more about the concept of this therapy, visit this link:
https://en.wikipedia.org/wiki/Cognitive_behavioral_therapy

Problem Scoping

CBT is a technique used by most therapists to cure patients out of stress and depression. But it has been observed that people do not wish to seek the help of a psychiatrist willingly. They try to avoid such interactions as much as possible. Thus, there is a need to bridge the gap between a person who needs help and the psychiatrist. Let us look at various factors around this problem through the 4Ws problem canvas.

Who Canvas – *Who has the problem?*

Who

Who are the stakeholders?	○ People who suffer from stress and are at the onset of depression.
What do we know about them?	○ People who are going through stress are reluctant to consult a psychiatrist.

What Canvas – *What is the nature of the problem?*

What

What is the problem?	○ People who need help are reluctant to consult a psychiatrist and hence live miserably.
How do you know it is a problem?	○ Studies around mental stress and depression available on various authentic sources.

Where Canvas – *Where does the problem arise?*

Where

What is the context/situation in which the stakeholders experience this problem?	○ When they are going through a stressful period of time ○ Due to some unpleasant experiences

Why Canvas – *Why do you think it is a problem worth solving?*

Why

What would be of key value to the stakeholders?	○ People get a platform where they can talk and vent out their feelings anonymously ○ People get a medium that can interact with them and applies primitive CBT on them and can suggest help whenever needed
How would it improve their situation?	○ People would be able to vent out their stress ○ They would consider going to a psychiatrist whenever required

Now that we have gone through all the factors around the problem, the problem statement templates go as follows:

Our	**People undergoing stress**	**Who?**
Have a problem of	Not being able to share their feelings	What?
While	They need help in venting out their emotions	Where?
An ideal solution would	Provide them a platform to share their thoughts anonymously and suggest help whenever required	Why

This leads us to the goal of our project which is:

"To create a chatbot which can interact with people, help them to vent out their feelings and take them through primitive CBT."

→ Data Acquisition

To understand the sentiments of people, we need to collect their conversational data so the machine can interpret the words that they use and understand their meaning. Such data can be collected from various means:

1. Surveys
2. Observing the therapist's sessions
3. Databases available on the internet
4. Interviews, etc.

→ Data Exploration

Once the textual data has been collected, it needs to be processed and cleaned so that an easier version can be sent to the machine. Thus, the text is normalised through various steps and is lowered to minimum vocabulary since the machine does not require grammatically correct statements but the essence of it.

→ Modelling

Once the text has been normalised, it is then fed to an NLP based AI model. Note that in NLP, modelling requires data pre-processing only after which the data is fed to the machine. Depending upon the type of chatbot we try to make, there are a lot of AI models available which help us build the foundation of our project.

→ Evaluation

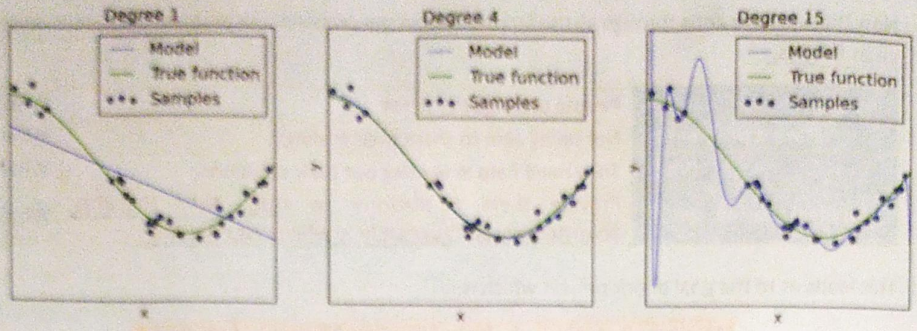
The model trained is then evaluated and the accuracy for the same is generated on the basis of the relevance of the answers which the machine gives to the user's responses. To understand the efficiency of the model, the suggested answers by the chatbot are compared to the actual answers.

Data Acquisition:-
→ Data is needed to train the model so that machine can interpret the words and understand its meanings. To understand sentiments of people we need conversational data that can be collected from surveys, online databases, therapist sessions, interviews etc.

Data Exploration:-
→ Data is processed and cleaned and a easier version is sent to the machine. Text is normalised through various steps and lowered to minimum vocabulary. The machine doesn't read grammatically correct data rather just the essence of it.

Modelling → once text is normalised it is fed into the machine. In NLP, modelling requires pre-processed data.

Evaluation → once the model is trained it is evaluated for accuracy on the basis of response the machine gives to user input. To understand the efficiency, the suggested answers are compared to the actual answers.



As you can see in the above diagram, the blue line talks about the model's output while the green one is the actual output along with the data samples.

Figure 1 The model's output does not match the true function at all. Hence the model is said to be underfitting and its accuracy is lower.

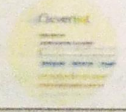
Figure 2 In the second one, the model's performance matches well with the true function which states that the model has optimum accuracy and the model is called a perfect fit.

Figure 3 In the third case, model performance is trying to cover all the data samples even if they are out of alignment to the true function. This model is said to be overfitting and this too has a lower accuracy.

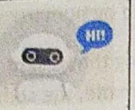
Once the model is evaluated thoroughly, it is then deployed in the form of an app which people can use easily.

Chatbots

As we have seen earlier, one of the most common applications of Natural Language Processing is a chatbot. There are a lot of chatbots available and many of them use the same approach as we used in the scenario above.. Let us try some of the chatbots and see how they work.

	• **Mitsuku Bot*** https://www.pandorabots.com/mitsuku/
	• **CleverBot*** https://www.cleverbot.com/
	• **Jabberwacky*** http://www.jabberwacky.com/
	• **Haptik*** https://haptik.ai/contact-us

* Images shown here are the property of individual organisations and are used here for reference purpose only.

	• **Rose*** http://ec2-54-215-197-164.us-west-1.compute.amazonaws.com/speech.php
	• **Ochatbot*** https://www.ometrics.com/blog/list-of-fun-chatbots/

Let us discuss!

- Which chatbot did you try? Name any one.
- What is the purpose of this chatbot?
- How was the interaction with the chatbot?
- Did the chat feel like talking to a human or a robot? Why do you think so?
- Do you feel that the chatbot has a certain personality?

As you interact with more and more chatbots, you would realise that some of them are scripted or in other words are traditional chatbots while others were AI-powered and had more knowledge. With the help of this experience, we can understand that there are 2 types of chatbots around us: Script-bot and Smart-bot. Let us understand what each of them mean in detail:

IMP

	Script-bot	Smart-bot
1	Script bots are easy to make	Smart-bots are flexible and powerful
2	Script bots work around a script which is programmed in them	Smart bots work on bigger databases and other resources directly
3	Mostly they are free and are easy to integrate to a messaging platform	Smart bots learn with more data
4	No or little language processing skills	Coding is required to take this up on board
5	Limited functionality	Wide functionality

The story speaker activity which was done in class 9 can be considered as a script-bot as in that activity we used to create a script around which the interactive story revolved. As soon as the machine got triggered by the person, it used to follow the script and answer accordingly. Other examples of script bot may include the bots which are deployed in the customer care section of various companies. Their job is to answer some basic queries that they are coded for and connect them to human executives once they are unable to handle the conversation.

On the other hand, all the assistants like Google Assistant, Alexa, Cortana, Siri, etc. can be taken as smart bots as not only can they handle the conversations but can also manage to do other tasks which makes them smarter.

Human Language VS Computer Language

Humans communicate through language which we process all the time. Our brain keeps on processing the sounds that it hears around itself and tries to make sense out of them all the time. Even in the classroom, as the teacher delivers the session, our brain is continuously processing everything and storing it in some place. Also, while this is happening, when your friend whispers something, the focus of your brain automatically shifts from the teacher's speech to your friend's conversation. So now, the brain is processing both the sounds but is prioritising the one on which our interest lies.

* Images shown here are the property of individual organisations and are used here for reference purpose only.

The sound reaches the brain through a long channel. As a person speaks, the sound travels from his mouth and goes to the listener's eardrum. The sound striking the eardrum is converted into neuron impulse, gets transported to the brain and then gets processed. After processing the signal, the brain gains understanding around the meaning of it. If it is clear, the signal gets stored. Otherwise, the listener asks for clarity to the speaker. This is how human languages are processed by humans.

On the other hand, the computer understands the language of numbers. Everything that is sent to the machine has to be converted to numbers. And while typing, if a single mistake is made, the computer throws an error and does not process that part. The communications made by the machines are very basic and simple.

Now, if we want the machine to understand our language, how should this happen? What are the possible difficulties a machine would face in processing natural language? Let us take a look at some of them here:

Arrangement of the words and meaning

There are rules in human language. There are nouns, verbs, adverbs, adjectives. A word can be a noun at one time and an adjective some other time. There are rules to provide structure to a language.

This is the issue related to the syntax of the language. Syntax refers to the grammatical structure of a sentence. When the structure is present, we can start interpreting the message. Now we also want to have the computer do this. One way to do this is to use the part-of-speech tagging. This allows the computer to identify the different parts of a speech.

Besides the matter of arrangement, there's also meaning behind the language we use. Human communication is complex. There are multiple characteristics of the human language that might be easy for a human to understand but extremely difficult for a computer to understand.

Analogy with programming language:

Different syntax, same semantics: 2+3 = 3+2

Here the way these statements are written is different, but their meanings are the same that is 5.

Different semantics, same syntax: 2/3 (Python 2.7) ≠ 2/3 (Python 3)

Here the statements written have the same syntax but their meanings are different. In Python 2.7, this statement would result in 1 while in Python 3, it would give an output of 1.5.

Think of some other examples of different syntax and same semantics and vice-versa.

Possible Difficulties in Processing NLP

1. Arrangement of words & meanings (word can be noun, verb, adjective)
2. Multiple Meanings of a word (context)
3. Perfect Syntax, No meaning. (Sentence can have perfect syntax but not meaning)

Multiple Meanings of a word

Let's consider these three sentences:

His face turned red after he found out that he took the wrong bag
What does this mean? Is he feeling ashamed because he took another person's bag instead of his? Is he feeling angry because he did not manage to steal the bag that he has been targeting?

The red car zoomed past his nose
Probably talking about the color of the car

His face turns red after consuming the medicine
Is he having an allergic reaction? Or is he not able to bear the taste of that medicine?

Here we can see that context is important. We understand a sentence almost intuitively, depending on our history of using the language, and the memories that have been built within. In all three sentences, the word red has been used in three different ways which according to the context of the statement changes its meaning completely. Thus, in natural language, it is important to understand that a word can have multiple meanings and the meanings fit into the statement according to the context of it.

Think of some other words which can have multiple meanings and use them in sentences.

Perfect Syntax, no Meaning

Sometimes, a statement can have a perfectly correct syntax but it does not mean anything. For example, take a look at this statement:

Chickens feed extravagantly while the moon drinks tea.

This statement is correct grammatically but does this make any sense? In Human language, a perfect balance of syntax and semantics is important for better understanding.

Think of some other sentences having correct syntax and incorrect semantics.

These are some of the challenges we might have to face if we try to teach computers how to understand and interact in human language. So how does Natural Language Processing do this magic?

Data Processing

Humans interact with each other very easily. For us, the natural languages that we use are so convenient that we speak them easily and understand them well too. But for computers, our languages are very complex. As you have already gone through some of the complications in human languages above, now it is time to see how Natural Language Processing makes it possible for the machines to understand and speak in the Natural Languages just like humans.

Since we all know that the language of computers is Numerical, the very first step that comes to our mind is to convert our language to numbers. This conversion takes a few steps to happen. The first step to it is Text Normalisation. Since human languages are complex, we need to first of all simplify them in order to make sure that the understanding becomes possible. Text Normalisation helps in cleaning up the textual data in such a way that it comes down to a level where its complexity is lower than the actual data. Let us go through Text Normalisation in detail.

Text Normalisation

In Text Normalisation, we undergo several steps to normalise the text to a lower level. Before we begin, we need to understand that in this section, we will be working on a collection of written text. That is, we will be working on text from multiple documents and the term used for the whole textual data from all the documents altogether is known as corpus. Not only would we go through all the steps of Text Normalisation, we would also work them out on a corpus. Let us take a look at the steps:

1. Sentence Segmentation

Under sentence segmentation, the whole corpus is divided into sentences. Each sentence is taken as a different data so now the whole corpus gets reduced to sentences.

Setence Segmentation
↓
Tokenisation
↓
Removing Stopwords, Special characters, Numbers
↓
Converting text to a common Case
↓
Stemming / Lemmatization.

In CBT, we learn to decipher the lies we are undermining ourselves with— based on the bias embedded in the things we say. For example, "I'm never going to make any friends" is an example of all-or-nothing thinking and we feel bad because we buy into this thought.

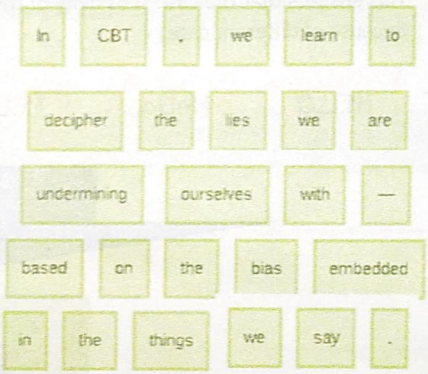
1. In CBT, we learn to decipher the lies we are undermining ourselves with— based on the bias embedded in the things we say.

2. For example, "I'm never going to make any friends" is an example of all-or-nothing thinking and we feel bad because we buy into this thought.

2 Tokenisation

After segmenting the sentences, each sentence is then further divided into tokens. Tokens is a term used for any word or number or special character occurring in a sentence. Under tokenisation, every word, number and special character is considered separately and each of them is now a separate token.

In CBT, we learn to decipher the lies we are undermining ourselves with— based on the bias embedded in the things we say.



3. Removing Stopwords, Special Characters and Numbers

In this step, the tokens which are not necessary are removed from the token list. What can be the possible words which we might not require?

Stopwords are the words which occur very frequently in the corpus but do not add any value to it. Humans use grammar to make their sentences meaningful for the other person to understand. But grammatical words do not add any essence to the information which is to be transmitted through the statement hence they come under stopwords. Some examples of stopwords are:

* Images shown here are the property of individual organisations and are used here for reference purpose only.

a, an, and, are, as, for, it, is, into, in, if, on, or, such, the, there, to, etc

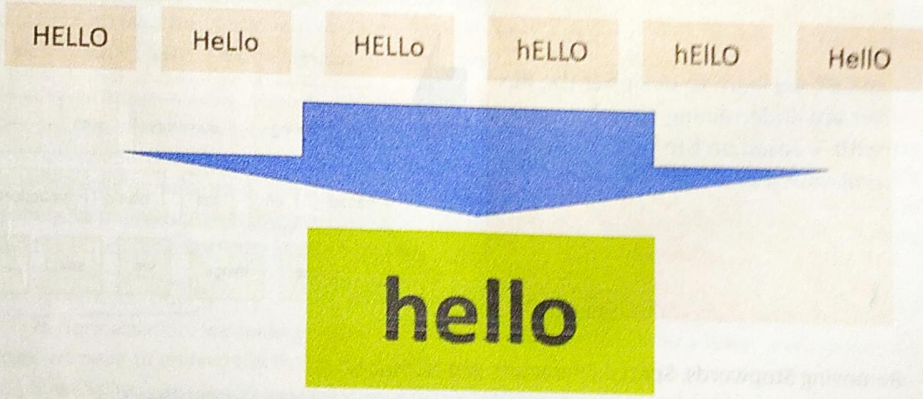
a	an	and	are	as	for
it	is	into	in	if	
on	or	such	the	there	to

These words occur the most in any given corpus but talk very little or nothing about the context or the meaning of it. Hence, to make it easier for the computer to focus on meaningful terms, these words are removed.

Along with these words, a lot of times our corpus might have special characters and/or numbers. Now it depends on the type of corpus that we are working on whether we should keep them in it or not. For example, if you are working on a document containing email IDs, then you might not want to remove the special characters and numbers whereas in some other textual data if these characters do not make sense, then you can remove them along with the stopwords.

4. Converting text to a common case

After the stopwords removal, we convert the whole text into a similar case, preferably lower case. This ensures that the case-sensitivity of the machine does not consider same words as different just because of different cases.



Here in this example, the all the 6 forms of hello would be converted to lower case and hence would be treated as the same word by the machine.

5. Stemming

In this step, the remaining words are reduced to their root words. In other words, stemming is the process in which the affixes of words are removed and the words are converted to their base form.

* Images shown here are the property of individual organisations and are used here for reference purpose only.

Word	Affixes	Stem
healed	-ed	heal
healing	-ing	heal
healer	-er	heal
studies	-es	studi
studying	-ing	study

Note that in stemming, the stemmed words (words which are we get after removing the affixes) might not be meaningful. Here in this example as you can see: healed, healing and healer all were reduced to heal but studies was reduced to studi after the affix removal which is not a meaningful word. Stemming does not take into account if the stemmed word is meaningful or not. It just removes the affixes hence it is faster.

6. Lemmatization

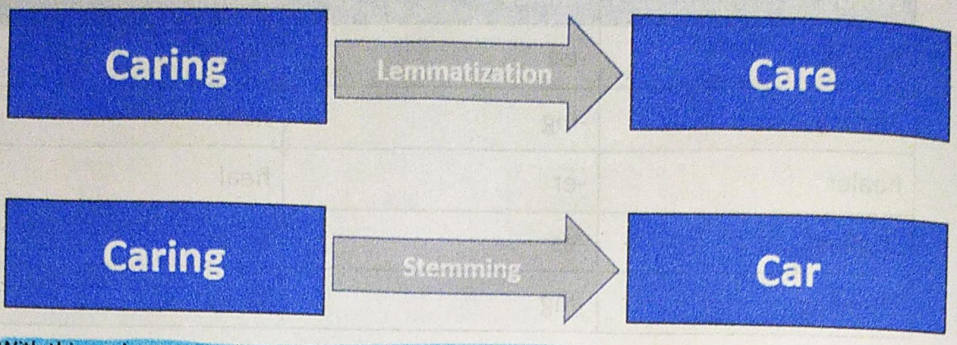
Stemming and lemmatization both are alternative processes to each other as the role of both the processes is same – removal of affixes. But the difference between both of them is that in lemmatization, the word we get after affix removal (also known as lemma) is a meaningful one. Lemmatization makes sure that lemma is a word with meaning and hence it takes a longer time to execute than stemming.

Word	Affixes	lemma
healed	-ed	heal
healing	-ing	heal
healer	-er	heal
studies	-es	study
studying	-ing	study

As you can see in the same example, the output for studies after affix removal has become study instead of studi.

* Images shown here are the property of individual organisations and are used here for reference purpose only.

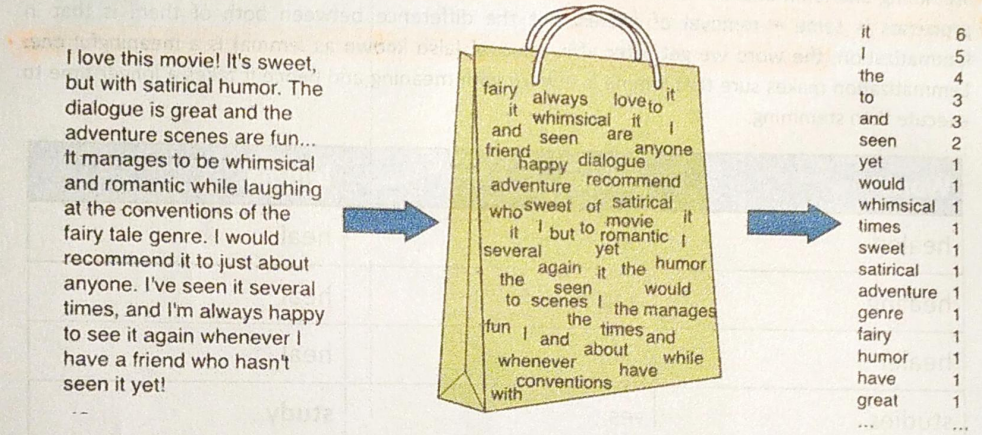
Difference between stemming and lemmatization can be summarized by this example:



With this we have normalised our text to tokens which are the simplest form of words present in the corpus. Now it is time to convert the tokens into numbers. For this, we would use the Bag of Words algorithm.

Bag of Words

Bag of Words is a Natural Language Processing model which helps in extracting features out of the text which can be helpful in machine learning algorithms. In bag of words, we get the occurrences of each word and construct the vocabulary for the corpus.



This image gives us a brief overview about how bag of words works. Let us assume that the text on the left in this image is the normalised corpus which we have got after going through all the steps of text processing. Now, as we put this text into the bag of words algorithm, the algorithm returns to us the unique words out of the corpus and their occurrences in it. As you can see at the right, it shows us a list of words appearing in the corpus and the numbers corresponding to it shows how many times the word has occurred in the text body. Thus, we can say that the bag of words gives us two things:

1. A vocabulary of words for the corpus
2. The frequency of these words (number of times it has occurred in the whole corpus).

Here calling this algorithm "bag" of words symbolises that the sequence of sentences or tokens does not matter in this case as all we need are the unique words and their frequency in it.

ALL

* Images shown here are the property of individual organisations and are used here for reference purpose only.

Here is the step-by-step approach to implement bag of words algorithm:

1. Text Normalisation: Collect data and pre-process it
2. Create Dictionary: Make a list of all the unique words occurring in the corpus. (Vocabulary)
3. Create document vectors: For each document in the corpus, find out how many times the word from the unique list of words has occurred.
4. Create document vectors for all the documents.

Let us go through all the steps with an example:

Step 1: Collecting data and pre-processing it. Text Normalisation

Document 1: *Aman and Anil are stressed*

Document 2: *Aman went to a therapist*

Document 3: *Anil went to download a health chatbot*

Here are three documents having one sentence each. After text normalisation, the text becomes:

Document 1: [aman, and, anil, are, stressed]

Document 2: [aman, went, to, a, therapist]

Document 3: [anil, went, to, download, a, health, chatbot]

Note that no tokens have been removed in the stopwords removal step. It is because we have very little data and since the frequency of all the words is almost the same, no word can be said to have lesser value than the other.

Step 2: Create Dictionary

Go through all the steps and create a dictionary i.e., list down all the words which occur in all three documents:

Dictionary:

aman	and	anil	are	stressed	went
download	health	chatbot	therapist	a	to

Note that even though some words are repeated in different documents, they are all written just once as while creating the dictionary, we create the list of unique words.

Step 3: Create document vector

In this step, the vocabulary is written in the top row. Now, for each word in the document, if it matches with the vocabulary, put a 1 under it. If the same word appears again, increment the previous value by 1. And if the word does not occur in that document, put a 0 under it.

aman	and	anil	are	stressed	went	to	a	therapist	download	health	chatbot
1	1	1	1	1	0	0	0	0	0	0	0

* Images shown here are the property of individual organisations and are used here for reference purpose only.

Since in the first document, we have words: aman, and, anil, are, stressed. So, all these words get a value of 1 and rest of the words get a 0 value.

Step 4: Repeat for all documents

Same exercise has to be done for all the documents. Hence, the table becomes:

aman	and	anil	are	stressed	went	to	a	therapist	download	health	chatbot
1	1	1	1	1	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1	1	0	0	0
0	0	1	0	0	1	1	1	0	1	1	1

In this table, the header row contains the vocabulary of the corpus and three rows correspond to three different documents. Take a look at this table and analyse the positioning of 0s and 1s in it.

Finally, this gives us the **document vector table** for our corpus. But the tokens have still not converted to numbers. This leads us to the final steps of our algorithm: TFIDF.

TFIDF: Term Frequency & Inverse Document Frequency

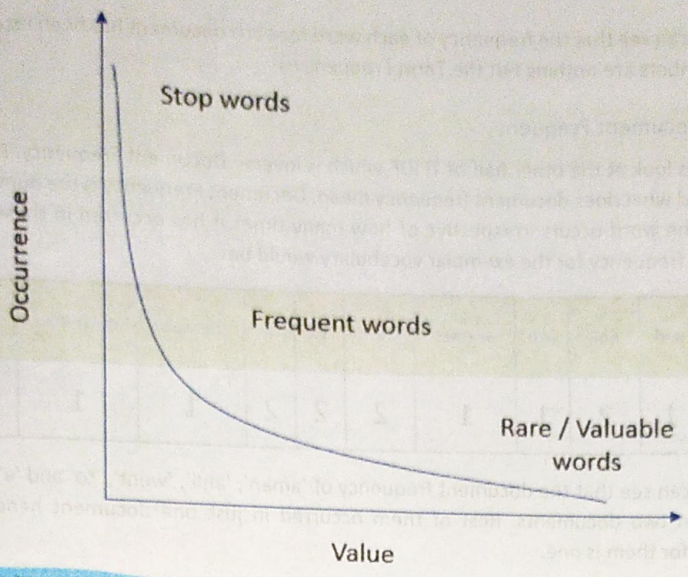
Suppose you have a book. Which characters or words do you think would occur the most in it?

Bag of words algorithm gives us the frequency of words in each document we have in our corpus. It gives us an idea that if the word is occurring more in a document, its value is more for that document. For example, if I have a document on air pollution, air and pollution would be the words which occur many times in it. And these words are valuable too as they give us some context around the document. But let us suppose we have 10 documents and all of them talk about different issues. One is on women empowerment, the other is on unemployment and so on. Do you think air and pollution would still be one of the most occurring words in the whole corpus? If not, then which words do you think would have the highest frequency in all of them?

And, this, is, the, etc. are the words which occur the most in almost all the documents. But these words do not talk about the corpus at all. Though they are important for humans as they make the statements understandable to us, for the machine they are a complete waste as they do not provide us with any information regarding the corpus. Hence, these are termed as stopwords and are mostly removed at the pre-processing stage only.

TFIDF - Term Frequency & Inverse Document Frequency

* Images shown here are the property of individual organisations and are used here for reference purpose only.



Take a look at this graph. It is a plot of occurrence of words versus their value. As you can see, if the words have highest occurrence in all the documents of the corpus, they are said to have negligible value hence they are termed as stop words. These words are mostly removed at the pre-processing stage only. Now as we move ahead from the stopwords, the occurrence level drops drastically and the words which have adequate occurrence in the corpus are said to have some amount of value and are termed as frequent words. These words mostly talk about the document's subject and their occurrence is adequate in the corpus. Then as the occurrence of words drops further, the value of such words rises. These words are termed as rare or valuable words. These words occur the least but add the most value to the corpus. Hence, when we look at the text, we take frequent and rare words into consideration.

Let us now demystify TFIDF. TFIDF stands for Term Frequency and Inverse Document Frequency. TFIDF helps un in identifying the value for each word. Let us understand each term one by one.

Term Frequency

Term frequency is the frequency of a word in one document. Term frequency can easily be found from the document vector table as in that table we mention the frequency of each word of the vocabulary in each document.

aman	and	anil	are	stressed	went	to	a	therapist	download	health	chatbot
1	1	1	1	1	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1	1	0	0	0
0	0	1	0	0	1	1	1	0	1	1	1

* Images shown here are the property of individual organisations and are used here for reference purpose only.

Here, you can see that the frequency of each word for each document has been recorded in the table. These numbers are nothing but the Term Frequencies!

Inverse Document Frequency

Now, let us look at the other half of TFIDF which is Inverse Document Frequency. For this, let us first understand what does document frequency mean. Document Frequency is the number of documents in which the word occurs irrespective of how many times it has occurred in those documents. The document frequency for the exemplar vocabulary would be:

aman	and	anil	are	stressed	went	to	a	therapist	download	health	Chatbot
2	1	2	1	1	2	2	2	1	1	1	1

Here, you can see that the document frequency of 'aman', 'anil', 'went', 'to' and 'a' is 2 as they have occurred in two documents. Rest of them occurred in just one document hence the document frequency for them is one.

Talking about inverse document frequency, we need to put the document frequency in the denominator while the total number of documents is the numerator. Here, the total number of documents are 3, hence inverse document frequency becomes:

aman	and	anil	are	stressed	went	to	a	therapist	download	health	chatbot
3/2	3/1	3/2	3/1	3/1	3/2	3/2	3/2	3/1	3/1	3/1	3/1

Finally, the formula of TFIDF for any word W becomes:

TFIDF(W) = TF(W) * log(IDF(W))

Here, log is to the base of 10. Don't worry! You don't need to calculate the log values by yourself. Simply use the log function in the calculator and find out!

Now, let's multiply the IDF values to the TF values. Note that the TF values are for each document while the IDF values are for the whole corpus. Hence, we need to multiply the IDF values to each row of the document vector table.

aman	and	anil	are	stress	went	to	a	therapist	download	health	chatbot
1*log(3/2)	1*log(3)	1*log(3/2)	1*log(3)	1*log(3)	0*log(3/2)	0*log(3/2)	0*log(3/2)	0*log(3)	0*log(3)	0*log(3)	0*log(3)
1*log(3/2)	0*log(3)	0*log(3/2)	0*log(3)	0*log(3)	1*log(3/2)	1*log(3/2)	1*log(3/2)	1*log(3)	0*log(3)	0*log(3)	0*log(3)
0*log(3/2)	0*log(3)	1*log(3/2)	0*log(3)	0*log(3)	1*log(3/2)	1*log(3/2)	1*log(3/2)	0*log(3)	1*log(3)	1*log(3)	1*log(3)

* Images shown here are the property of individual organisations and are used here for reference purpose only.

Here, you can see that the IDF values for Aman in each row is the same and similar pattern is followed for all the words of the vocabulary. After calculating all the values, we get:

aman	and	anil	are	stress	went	to	a	therapist	download	health	chatbot
0.176	0.477	0.176	0.477	0.477	0	0	0	0	0	0	0
0.176	0	0	0	0	0.176	0.176	0.176	0.477	0	0	0
0	0	0.176	0	0	0.176	0.176	0.176	0	0.477	0.477	0.477

Finally, the words have been converted to numbers. These numbers are the values of each for each document. Here, you can see that since we have less amount of data, words like 'are' and 'and' also have a high value. But as the IDF value increases, the value of that word decreases. That is, for example:

Total Number of documents: 10

Number of documents in which 'and' occurs: 10

Therefore, IDF(and) = 10/10 = 1

Which means: log(1) = 0. Hence, the value of 'and' becomes 0.

On the other hand, number of documents in which 'pollution' occurs: 3

IDF(pollution) = 10/3 = 3.3333...

Which means: log(3.3333) = 0.522; which shows that the word 'pollution' has considerable value in the corpus.

Summarising the concept, we can say that:

1. Words that occur in all the documents with high term frequencies have the least values and are considered to be the stopwords.
2. For a word to have high TFIDF value, the word needs to have a high term frequency but less document frequency which shows that the word is important for one document but is not a common word for all documents.
3. These values help the computer understand which words are to be considered while processing the natural language. The higher the value, the more important the word is for a given corpus.

Step-1 Document Vectors (Term Frequency table)

Step-2 Document Frequency Table

Step-3 inverse Document Frequency table

Step-4 TFIFD Table

$TFIFD(w) = TF(w) \times \log(IDF(w))$

* Imag d here for reference purpose only.

Applications of TFIDF

TFIDF is commonly used in the Natural Language Processing domain. Some of its applications are:

Document Classification	Topic Modelling	Information Retrieval System	Stop word filtering
Helps in classifying the type and genre of a document.	It helps in predicting the topic for a corpus.	To extract the important information out of a corpus.	Helps in removing the unnecessary words out of a text body.

DIY – Do It Yourself!

Here is a corpus for you to challenge yourself with the given tasks. Use the knowledge you have gained in the above sections and try completing the whole exercise by yourself.

The Corpus

Document 1: We can use health chatbots for treating stress.

Document 2: We can use NLP to create chatbots and we will be making health chatbots now!

Document 3: Health Chatbots cannot replace human counsellors now. Yay >< !! @1nteLA!4Y

Accomplish the following challenges on the basis of the corpus given above. You can use the tools available online for these challenges. Link for each tool is given below:

1. Sentence Segmentation: https://tinyurl.com/y36hd92n
2. Tokenisation: https://text-processing.com/demo/tokenize/
3. Stopwords removal: https://demos.datasciencedojo.com/demo/stopwords/
4. Lowercase conversion: https://caseconverter.com/
5. Stemming: http://textanalysisonline.com/nltk-porter-stemmer
6. Lemmatisation: http://textanalysisonline.com/spacy-word-lemmatize
7. Bag of Words: Create a document vector table for all documents.
8. Generate TFIDF values for all the words.
9. Find the words having highest value.
10. Find the words having the least value.

Evaluation

Introduction

Till now we have learnt about the 4 stages of AI project cycle, viz. Problem scoping, Data acquisition, Data exploration and modelling. While in modelling we can make different types of models, how do we check if one's better than the other? That's where Evaluation comes into play. In the Evaluation stage, we will explore different methods of evaluating an AI model. Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future

What is evaluation?

Evaluation is the process of understanding the reliability of any AI model, based on outputs by feeding test dataset into the model and comparing with actual answers. There can be different Evaluation techniques, depending of the type and purpose of the model. Remember that It's not recommended to use the data we used to build the model to evaluate it. This is because our model will simply remember the whole training set, and will therefore always predict the correct label for any point in the training set. This is known as overfitting.

Firstly, let us go through various terms which are very important to the evaluation process.

Model Evaluation Terminologies

There are various new terminologies which come into the picture when we work on evaluating our model. Let's explore them with an example of the Forest fire scenario.

The Scenario

Imagine that you have come up with an AI based prediction model which has been deployed in a forest which is prone to forest fires. Now, the objective of the model is to predict whether a forest fire has broken out in the forest or not. Now, to understand the efficiency of this model, we need to check if the predictions which it makes are correct or not. Thus, there exist two conditions which we need to ponder upon: Prediction and Reality. The prediction is the output which is given by the machine and the reality is the real scenario in the forest when the prediction has been made. Now let us look at various combinations that we can have with these two conditions.

Case 1: Is there a forest fire?

Prediction: Yes

Reality: Yes

True Positive

Here, we can see in the picture that a forest fire has broken out in the forest. The model predicts a Yes which means there is a forest fire. The Prediction matches with the Reality. Hence, this condition is termed as **True Positive**.

Case 2: Is there a forest fire?

Prediction: No

Reality: No

True Negative

Here there is no fire in the forest hence the reality is No. In this case, the machine too has predicted it correctly as a No. Therefore, this condition is termed as **True Negative**.

* Images shown here are the property of individual organisations and are used here for reference purpose only.

Case 3: Is there a forest fire?

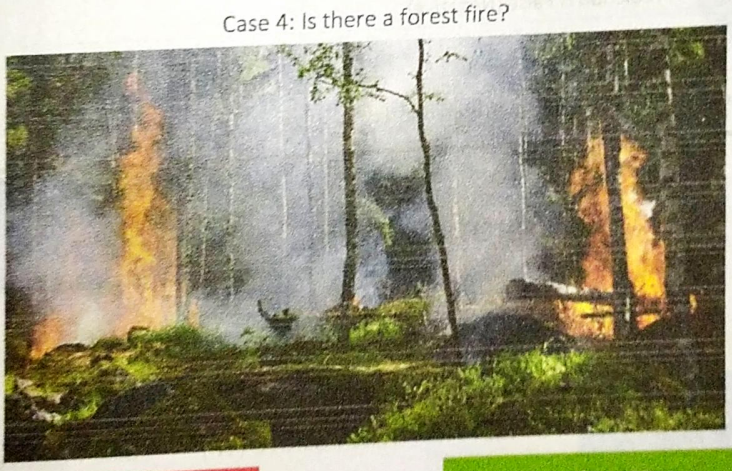
Prediction: Yes

Reality: No

False Positive

Here the reality is that there is no forest fire. But the machine has incorrectly predicted that there is a forest fire. This case is termed as **False Positive**.

Case 4: Is there a forest fire?



Prediction: No

Reality: Yes

False Negative

* Images shown here are the property of individual organisations and are used here for reference purpose only.

Here, a forest fire has broken out in the forest because of which the Reality is Yes but the machine has incorrectly predicted it as a No which means the machine predicts that there is no Forest Fire. Therefore, this case becomes **False Negative**.

Confusion matrix

The result of comparison between the prediction and reality can be recorded in what we call the confusion matrix. The confusion matrix allows us to understand the prediction results. Note that it is not an evaluation metric but a record which can help in evaluation. Let us once again take a look at the four conditions that we went through in the Forest Fire example:

TP
- True Positive
- Prediction & Reality matches (True)
- Prediction is True (Positive)

TN
- True Negative
- Prediction & Reality matches (True)
- Prediction is False (Negative)

FP
- False Postive
- Prediction and Reality do not match (False)
- Prediction is True (Positive)

FN
- False Negative
- Prediction and Reality do not match (False)
- Prediction is False (Negative)

Let us now take a look at the confusion matrix:

The Confusion Matrix		Reality	
		Yes	No
Prediction	Yes	True Positive (TP)	False Positive (FP)
	No	False Negative (FN)	True Negative (TN)

Prediction and Reality can be easily mapped together with the help of this confusion matrix.

* Images shown here are the property of individual organisations and are used here for reference purpose only.

Evaluation Methods

Now as we have gone through all the possible combinations of Prediction and Reality, let us see how we can use these conditions to evaluate the model.

Accuracy

Accuracy is defined as the percentage of correct predictions out of all the observations. A prediction can be said to be correct if it matches the reality. Here, we have two conditions in which the Prediction matches with the Reality: True Positive and True Negative. Hence, the formula for Accuracy becomes:

$$Accuracy = \frac{Correct\ prediction}{Total\ cases} * 100\%$$

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} * 100\%$$

Here, total observations cover all the possible cases of prediction that can be True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN).

As we can see, Accuracy talks about how true the predictions are by any model. Let us ponder:

Is high accuracy equivalent to good performance?

_____

_____

How much percentage of accuracy is reasonable to show good performance?

_____

_____

Let us go back to the Forest Fire example. Assume that the model always predicts that there is no fire. But in reality, there is a 2% chance of forest fire breaking out. In this case, for 98 cases, the model will be right but for those 2 cases in which there was a forest fire, then too the model predicted no fire.

Here,

True Positives = 0

True Negatives = 98

Total cases = 100

Therefore, accuracy becomes: (98 + 0) / 100 = 98%

Accuracy = Correct Predictions / Total Cases × 100

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

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

F1 Score = (Precision × Recall) / (Precision + Recall)

* Images shown here are the property of individual organisations and are used here fo

Prediction: Always No

Reality: 2% probability of Yes

98% accurate
But is it usable?

This is a fairly high accuracy for an AI model. But this parameter is useless for us as the actual cases where the fire broke out are not taken into account. Hence, there is a need to look at another parameter which takes account of such cases as well.

Precision

Precision is defined as the percentage of true positive cases versus all the cases where the prediction is true. That is, it takes into account the True Positives and False Positives.

$$Precision = \frac{True\ Positive}{All\ Predicted\ Positives} * 100\%$$

$$Precision = \frac{TP}{TP + FP} * 100\%$$

Going back to the Forest Fire example, in this case, assume that the model always predicts that there is a forest fire irrespective of the reality. In this case, all the Positive conditions would be taken into account that is, True Positive (Prediction = Yes and Reality = Yes) and False Positive (Prediction = Yes and Reality = No). In this case, the firefighters will check for the fire all the time to see if the alarm was True or False.

You might recall the story of the boy who falsely cries out that there are wolves every time and so when they actually arrive, no one comes to his rescue. Similarly, here if the Precision is low (which means there are more False alarms than the actual ones) then the firefighters would get complacent and might not go and check every time considering it could be a false alarm.

* Images shown here are the property of individual organisations and are used here for reference purpose only.

Think of some more examples having:

- High False Negative cost

- High False Positive cost

Both measures are important

High Precision,

Precision = (TP) / (TP + FP)

High Recall,

Recall = (TP) / (TP + FN)

We need something that account for the 2 metrics

To conclude the argument, we must say that if we want to know if our model's performance is good, we need these two measures: Recall and Precision. For some cases, you might have a High Precision but Low Recall or Low Precision but High Recall. But since both the measures are important, there is a need of a parameter which takes both Precision and Recall into account.

F1 Score

F1 score can be defined as the measure of balance between precision and recall.

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

Take a look at the formula and think of when can we get a perfect F1 score?

An ideal situation would be when we have a value of 1 (that is 100%) for both Precision and Recall. In that case, the F1 score would also be an ideal 1 (100%). It is known as the perfect value for F1 Score. As the values of both Precision and Recall ranges from 0 to 1, the F1 score also ranges from 0 to 1.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$Recall = \frac{TP}{TP + FN}$$

Now as we notice, we can see that the Numerator in both Precision and Recall is the same: True Positives. But in the denominator, Precision counts the False Positives while Recall takes False Negatives into consideration.

Let us ponder... Which one do you think is better? Precision or Recall? Why?

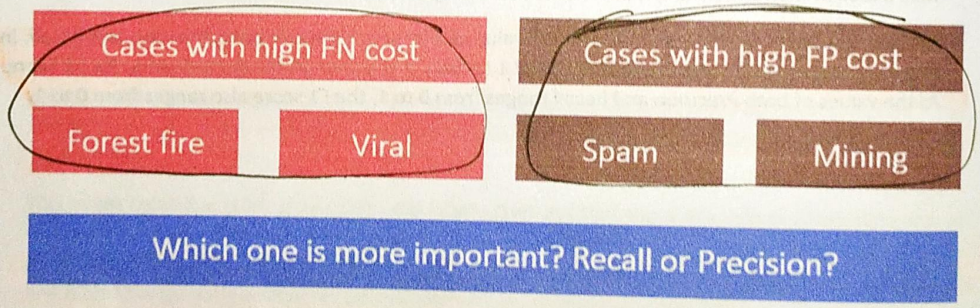
Which Metric is Important?

Choosing between Precision and Recall depends on the condition in which the model has been deployed. In a case like Forest Fire, a False Negative can cost us a lot and is risky too. Imagine no alert being given even when there is a Forest Fire. The whole forest might burn down.

Another case where a False Negative can be dangerous is Viral Outbreak. Imagine a deadly virus has started spreading and the model which is supposed to predict a viral outbreak does not detect it. The virus might spread widely and infect a lot of people.

On the other hand, there can be cases in which the False Positive condition costs us more than False Negatives. One such case is Mining. Imagine a model telling you that there exists treasure at a point and you keep on digging there but it turns out that it is a false alarm. Here, False Positive case (predicting there is treasure but there is no treasure) can be very costly.

Similarly, let's consider a model that predicts that a mail is spam or not. If the model always predicts that the mail is spam, people would not look at it and eventually might lose important information. Here also False Positive condition (Predicting the mail as spam while the mail is not spam) would have a high cost.



This makes Precision an important evaluation criteria. If Precision is high, this means the True Positive cases are more, giving lesser False alarms.

But again, is good Precision equivalent to a good model performance? Why?

Prediction: 10 cases of TP

Reality: 20 cases of yes

100% precise
But is it usable?

IMP

Let us consider that a model has 100% precision. Which means that whenever the machine says there's a fire, there is actually a fire (True Positive). In the same model, there can be a rare exceptional case where there was actual fire but the system could not detect it. This is the case of a False Negative condition. But the precision value would not be affected by it because it does not take FN into account. Is precision then a good parameter for model performance?

Recall

Another parameter for evaluating the model's performance is Recall. It can be defined as the fraction of positive cases that are correctly identified. It majorly takes into account the true reality cases where in Reality there was a fire but the machine either detected it correctly or it didn't. That is, it considers True Positives (There was a forest fire in reality and the model predicted a forest fire) and False Negatives (There was a forest fire and the model didn't predict it).

* Images shown here are the property of individual organisations and are used here for reference purpose only.

Let us explore the variations we can have in the F1 Score:

Precision	Recall	F1 Score
Low	Low	Low
Low	High	Low
High	Low	Low
High	High	High

In conclusion, we can say that a model has good performance if the F1 Score for that model is high.

Let's practice!

Let us understand the evaluation parameters with the help of examples.

Challenge

Find out Accuracy, Precision, Recall and F1 Score for the given problems.

Scenario 1:

In schools, a lot of times it happens that there is no water to drink. At a few places, cases of water shortage in schools are very common and prominent. Hence, an AI model is designed to predict if there is going to be a water shortage in the school in the near future or not. The confusion matrix for the same is:

The Confusion Matrix	Reality: 1	Reality: 0
Predicted: 1	22	12
Predicted: 0	47	118

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} = \frac{22+118}{22+118+12+47} = \frac{140}{199} = 0.70$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{22}{22+12} = \frac{22 \times 11}{34 \times 17} = 0.64$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{22}{22+47} = \frac{22}{69} = 0.31$$

$$\text{F1 Score} = \frac{P \times R}{P+R} = \frac{0.1984}{0.95} = 0.20$$

Scenario 2:

Nowadays, the problem of floods has worsened in some parts of the country. Not only does it damage the whole place but it also forces people to move out of their homes and relocate. To address this issue, an AI model has been created which can predict if there is a chance of floods or not. The confusion matrix for the same is:

The Confusion Matrix	Actual: 1	Actual: 0
Predicted: 1	0	3
Predicted: 0	3	94

Scenario 3:

A lot of times people face the problem of sudden downpour. People wash clothes and put them out to dry but due to unexpected rain, their work gets wasted. Thus, an AI model has been created which predicts if there will be rain or not. The confusion matrix for the same is:

The Confusion Matrix	Actual: 1	Actual: 0
Predicted: 1	5	0
Predicted: 0	45	50

Scenario 4:

Traffic Jams have become a common part of our lives nowadays. Living in an urban area means you have to face traffic each and every time you get out on the road. Mostly, school students opt for buses to go to school. Many times the bus gets late due to such jams and students are not able to reach their school on time. Thus, an AI model is created to predict explicitly if there would be a traffic jam on their way to school or not. The confusion matrix for the same is:

The Confusion Matrix	Actual: 1	Actual: 0
Predicted: 1	50	50
Predicted: 0	0	0