Python and AI Power-Up Program Offline Class-20250820_190816-Meeting Recording

August 20, 2025, 1:38PM

2h 31m 22s

Margi Varmora started transcription

Tarun 0:07

OK, so whatever is right, it's from both recap notebook and as well as the oops notebook oops and decorator. So probably if you have checked the code base you have two notebooks right inside the notebook folder.

So first we'll take overall quiz, then we'll talk about the pyrantic one, how you can add that custom conditions. So once that is done, then we'll go with NLP and have few slides, right? So yeah, so let's get started. Can anyone tell me the output of this?

So we how many print statements do we have? Just one.

This will be empty.

We took V1 will be at the list.

So this is for data class and as well as state persistence. So what about the second one?

I.

One, one, one. Send this to one comma 2.

Yeah, empty. But it's it's just and no, it will be one comma two. This is 3. Why? Because I'm passing empty.

Yeah.

No, no, the because why you say why is a parameter? It's a list word.

No, I understand that the memory address of Y is still the same. So after this should be add print add four. Would it be 124 or four? One to two print add add 4.

So it will be 3/4.

OK, here you cleared it. The Y is taking this persistence. Can we clean the Y? Can we clean Y? Which one? Can we clean Y directly here outside? Yeah.

It is an error, right? Because this is global. Sorry, this is local and this is so every time we call by, isn't it reassigned to MPA? It is not reassigning into MPA, it is doing the reference of the MPA.

So when you are not passing it, it is using the old reference. So it like old reference.

So old reference is 1 comma two. So here it's empty the encapsulation but.

Now here for this example right?
I'll just tell one thing. Collab doesn't run threading. OK, so once you start something it will keep on running right? So giving this context, can anyone tell me the output of this?
Yes, both thread started, then thread A started, thread B started. Yeah, then thread DA started.
Thread A started then thread B started then both threads started. No, no, no, no, no, no. The both threads started then thread A and thread B. I need to say threads. No, no, both threads started will be after the starting.
In both the threads OK and then at the end will be the the thread A finish and thread B finished. Huh. But we are using collab now. I said there is no join, correct? There is no join. If I run this, it's thread A starting, thread B starting and both threads running if I run the same code.
In VS code. Now can you tell me the output? Yes, then there will be an empty. What I'll do is I'll run terminal because I have the output there CD desktop.
CD Python workshop, CD scripts, CD Python recap.
Yeah, so can anyone tell me the two comments? I want to activate the environment variable, So what is the command for that?
First, what is the first keyword?
Source then OK auto complete PPNV bin activate. So this PPNV can be anything. OK how do I create the environment variable?
I mean newly creation. No, that is I'm activating. Let's suppose there is no PPNV. How do I create this PPNV?
Python 3 hyphen M then.
So here I'll just run quiz dot PY. So thread A starting, thread B starting, both threads started, thread B finished, thread A finished. So there is no order. If I remove this comma then.
What?
It's already saved.
OK.
OK.
So I removed the comma from here.
OK, why is it working?
It is not saved, I guess it is saved.
OK, let me run it here.

I have lost my name somewhere.
I'll use this example that we took yesterday. So this is the download example.
So I have two images files and then I'm running threading here instead of URL I'll remove this comma.
Uh, here I'm getting the error.
If I add comma, it's working, so let's reuse the same example.
I have worker right? I will copy this.
And here instead of download image what will I do?
I will just add worker and here I'll just give A comma B.
And this is a name.
And I'll remove the comma.
OK, worker is printing.
OK, this is some weird. So usually you have to give comma. That's the basic syntax of using threading. So threading dot thread. What is the first parameter?
Function name which is target and then you have ERGS which is arguments. Let's suppose if I'm using worker which is a function. If that has any arguments then I have to define ERGS and after ERGS you have to give comma. That is a standard template or syntax.
And never run threading, multiprocessing, asynchronous programming on collab, right? So if you're running asynchronous programming on collab, you have these two statements import NEST.
There is a library called Nest Asincio and you just have to apply this Nest essential apply. This is only for async related programming language, right? So you have async, you define your function and then you're using await.
So that will only work when you apply it. If not, it will not work on collab. That's the reason why we had to use VS code only for those examples. OK, so this is the final example. Can anyone tell me the output of this?
So here if you notice the order first is B. Hello from B.
Is this clear? Yes, OK.
So now what we can do is let's just relook into the previous examples that we had on how to add custom logic for your validation. So basically field validator that we had is deprecated that we were trying yesterday. There were also added one more functionality which is called.
Model validator, right? So there are two possibilities. One is either you can use root validator.

Or field validator.
If you are using Pydantic version one.
But in Pydantic version too.
They're mainly using model validator.
So if we have to test that, what we can do is we can do import pydantic.
And then just run pydantic dot under score under score version.
And now if I run this it will show 2.11.7.
OK, so we are using version two and what are we supposed to define? Model validator. So what is the syntax of defining pydantic?
Class name. Then what is the first keyword base model? And once we define base model, what is the next thing?
You have to define attributes, right? So this attributes will have its own validation. For example, I have name and how do I define any attribute? If I have my own custom variable, I have to define field. So what is this three dots?
No default values and then you have minimum length. For example in name I need it to be at least three and maximum is 35. Only this is something that I will allow and then age it should be greater than 16 and now here we have country then e-mail. So basically if you are using e-mail.
TR you need to install a package pip install pydantic then in brackets you have e-mail. If not we can directly define pattern which is regex and here what we are trying to do now is I'll show 2 syntax in pydantic one is.
Once you define any variable, what is the after effect like? What are the? Let's suppose I have two attributes that is already defined. Using these two attributes, what is the next validation I need to define that is added in mode equals to after.
OK, so mode equals to after in the sense. Here you can define any logic that you need. So this logic is defined for can drink the example that we had. So now can drink is taking two variables, one is country and one more is age.
So now if you see here, every time you define any function under class, it's called method right? And whenever we define a method, what is the first argument we need to use self. So this is a method.
I'm using self whether it is pydantic or data classes or normal class methods will have self as its first argument and here now if you see self dot country if user enters USA everything in capital, I'm just checking for the condition if it is UK.
And if it is more than 18, you want to have this can drink as true. And same goes for other two arguments that I have USA and India. If the age is more than 21 then true.

And if in case you want to add more countries, that logic can be added. But here I'm just using it for three and what is the mode it's after. So now what will happen is when you instantiate this object or create the object of this, you will only define name, age, country and e-mail.
If you directly want to assign it as scanned ring, obviously this is something that you will not expose to the user. You will tell user only to enter these four things and the fifth thing will be generated automatically using your logic that you have.
So here I'm defining Tarun, then I am defining 17 just to test if can drink is working, then country India and then you have e-mail as dummy.
And now if you see it is false, this scan drink was not defined if I give this as 21.
It is true and if I remove this at and if I add hashtag will this work?
So it will give validation error.
This is for mode after, so I also tried with root validator, but as I said root validator is for version one.
So this will not work.
Now here we have one more example for before. So what I'm trying to do is I will just define a class and I have two variables. So here what I need to give control is as soon as user gives any.
Entry point for the attributes that we have, we have to put validation directly. So if you see here there is no field, I'm defining no field. I want to have my own custom validation here. What was your custom validation? You're checking it based on a pattern.
You're checking it based on certain condition greater than 16. So for that you used a field. Here the goal is to have custom validation and defining mode validator. What is the mode? It's before.
Right, I want to check before itself while I'm defining that variable, not after I define that variable. So here if you see I'm checking if it is a valid employee e-mail. If it ends with atyantik.com then it's fine. If it doesn't end, you have to print invalid employee not registered.
So now what you're trying to do is you're not checking for the pattern of the e-mail, you're just checking if it ends with atyantic.com or not. So now if I run this.
And if I define any name, if you see here it ends with that particular URL, I'll just remove this.
And this one works right? But what if I just define?
This is a valid e-mail right? Dummy. Probably I can have my own domain as

dummy.com tarun@dummy.com. If I run this it's an error. So you have the same message invalid employee not registered and then you have that particular error message.
So as I mentioned earlier, whenever it comes to pedantic, commonly you will encounter validation error or.
Parsing error from Jason. Is this clear? Mod model validator?
Huh. This is validation error, right? I don't have this. This pattern doesn't match.
So basically OK, so this is this is a rise function.
And use validation error even though.
OK, so this particular thing is there, right? Let's suppose you're defining any functions. So if you want to have your own meaningful error message, we use a function called raise.
So let's suppose.
Everyone knows try and accept.
So in try and accept what we will do is I'm asking a equals to 10 and then I will make a error message called print a.
Under score 0. So this should probably give one error.
Oh.
So now I have accept here. What I will do is I'll just I'll raise. Hey, you can't do this.
So if you see you have.
So it is not an exception, it is an error. So that's why if you write as space error. Oh one second, one second. If I define this thing.
Let's do the same. Java is exception and error. Except this error. Except that error.
Java has two classes, exception and error. Exception can be managed directly, but error.
Uh, memory out of bound is an error. It's not an Excel. OK.
Return E.
They look great.
Error 0 division error is not inherited from exception. It is inherited from error. Mm-hmm. So if we try different error message, let's suppose.
10:20.
And now here if I do a off.
Three, which is a key error.
I didn't accept them now.
Yes.

In the top.
OK, I'll get to this example, but usually whenever it comes to validation error.
You can place the exception from from which one, yeah.
Where were I? Where did that code go now?
OK.
I'll probably remote it. OK, fine.
No, Control Z doesn't work on now. OK, search.
Right.
That will show other code, but that's fine.
Hero Divide.
Atiller, you understood the validation error. OK, so while I was going through the document, I also found one more segment where pydantic is very useful. So that one is Jason's schema. So what are the three things that we said pydantic is useful for? Can anyone recall?
Parsing. First one is for parsing which is for Jason with LLM. Second validation validation.
Where you are passing certain checks and telling.
Whether it's a right entry or not.
And third thing is Jason parsing. Jason parsing is added here that is for defining schema, right? So defining schema. So here I will take one example where we can cover both schema and as well as.
Jason parsing, right? So same customer example. So we have this customer, right? I will copy this.
I will paste it here.
So there is a function called customer dot model Jason schema. So if you just run this you will get this schema of entire entry that you're supposed to give. So you have properties name, the name should be Max length of 35, minimum length is 3, title is name and then type is string.
So this will repeat for other attributes as well. So now this is the model schema. So once you define model schema, what I'm trying to do is I'm defining an object. So here the object is customer, Tarun age, whatever parameters are required. So how many parameters do we have?
Attributes 123 and four. I need to only define four, but if you see you also have can drink, but can drink is added as any of. That means it's used for optional and now I'll just define customer one.

And once you define customer one, you just have to use a function called model dump Jason.
Schema is to get this part and if you do model Jason model dump Jason, it's for a object. This is for the entire class, right?
And now if I do model dum Jason, whatever output I have, it's in Jason.
And that you can directly load in Jason.
Import Jason.
And now if you do this, it's in proper Jason format. So whatever you have here, if you notice this thing, it's still in string. So if you want to get in a proper Jason format, we can just import Jason, then Jason dot loads that particular same line.
So these are the two important functions. One is to get schema you can use model Jason schema and for once you define any object and if you want to convert that into Jason, you can just use model Jason. Now how can we validate with that?
No, no, this we are doing today only. OK, so.
Yeah, from the schema, we have the schema of the Jason. Yeah, right. Now we want to validate some Jason value with the schema.
A direct check is in order in pydantic. Jason just has two functions, one is Jason dot dump and one more is loads. So one is load and loads. It's the same thing. Then you have dump and dumps.
And this is the error message that you will usually get from Pydantic. I told it one is validation error error and 2nd is this Jason decode error. These are the two common issues from Pydantic.
Now if you notice this thing right this function model dum Jason, I'll show one live example.
Gemma 3 use cases.
So do you recall this example that I showed yesterday? So I have a file which is a invoice. So from this invoice I need to extract description, quantity, unit price, total price of a line item. So this is in one class.
And what is this class?
This one which is 500 and then I need to also extract the main details. In this main details you have vendor name, vendor address, vendor e-mail, client name, client address. These line items is all these details.
OK, so now when you run the agent.
If you see the agent, it is. What is this?
Can anyone recall what this is?

How does REPR works? You have the class name and all the attributes name. So here as soon as you get the response back from an agent, what is the agent trying to do? I'll just tell that in simple line. So I'm defining.
Agent. This agent is supposed to response back in a Jason, but what am I passing here?
That is the class. What invoice data is a plus plus by so agent needs to return the response back in.
Class which is pydantic so that you can extract the Jason. So here if you notice you have model dump which will extract you the entire Jason. So this is again one of the function from pydantic.
So is this clear? The agent give it in Jason format and then convert to the class so that is not inconsistent method. So Jason can give you this. I mean agent can generate Jason file but usually what it will do is.
Let's suppose I'm using an agent and I'm telling.
Agent dot run some prompt. OK so this should be response. So this agent you are giving certain instructions saying hey you have to generate in a Jason file. What this agent will do now is it will tell here is.
Your response in Jason and then it will write Jason, Jason, Jason.
This will cause an error because this you can't directly parse in Jason, right? So this text itself should not come.
And even if this won't come right, if you give some strict rules, this won't come sometimes.
You have this uh response model equal to his voice data. Huh. It will not be response model as Jason.
This one? Yeah, no, that won't work. So usually whenever we want to generate Jason, your class has to be in pydantic.
That will just ensure. Let's suppose what was the purpose of defining. This is a key and the goal of an agent is to get the value keys defined and in order to get the value it has to have some context what that key is about. So that is where we are defining this description.
So now based on this key and description, it will generate a value. What we want to ensure is agent should not generate anything apart from a Jason file, right? So what you will usually notice if you don't define pydentic is it will add one line. Here is your final Jason.
Even if you do some strict prompting, it will give you markdown. OK, So what it will

do is it will at least give you this markdown, but this will be missing sometimes.
So you can try that inside GPT also if you give some prompt and if you hide something in triple quotes it will skip this.
So 90% of the time it will work, but the 10% of the time it's inconsistent. So just to avoid those inconsistency, every single frameworks they rely on pydantic to generate the response in Jason. So that is just to avoid inconsistency.
That is internal.
Huh. When it comes to pythantic, huh. So if you see this, it is.
You know, based on which context of identity like it knows how to just this function, this particular format.
Most of them support this. Every single framework supports Pytantic. Open AI itself started this. Yeah, I mean not Pytantic, the Jason parsing the first release. How to generate structured responses? So what should be structured?
Responses I think. So this was the first release OK which open AI had done. So if I open this example, So what is this now?
So these folks itself released the first version of how you can get the structured response. So as soon as they used Pyrantic, everyone started using Pyrantic to build structured responses.
What do we need to give Jason True?
What in the example that OK, this is the PIA frameworks logic, so usually by default you have markdown to be true.
So if I show other examples, so here I'm building a translate agent. Here you have markdown equals to markdown equals to true. So that is how I have no built their framework. It's like if you need Jason you can use this argument. So but if you're passing the identic.
Class then whatever you pass here won't matter. So this is just for them to add new parameter. OK, so usually even this should generate you a Jason file, right? But what these folks have done is I've added this additional keyword.
So if you notice one more thing here, you have role, you have system message. Basically these two things are same.
So if you want to define a system prompt, you define a role play, you define instruction, and then you define if you need any output format. But if you see instructions and role play again are different.
So these two things are not at all required. You can pass this to inside a system prompt, but still they wrote it separately. That is how they define their framework.

But basically if you need just structured response and if you're building your own framework.
You just need pedantic and a LLM call. That's it.
OK. Is this clear? Yes.
All right, we'll get inside the NLP. So before we can get started with NLP again, we have some disclaimers. So for NLP rag agents and fine tuning session, we'll go in a different approach. So as of now what we did was we were looking into new, new concept every single day and we had some examples.
But in this concept right you'll have same concept and you will have to apply it in a project. So if today if we learn some concept tomorrow, how can you use this concept in a existing examples right project based approach. So for these three things.
Four things NLP, rag, agents and fine tuning will go on a project based approach. We learn a concept, we replicate that in a existing project and we'll go with the template approach itself. I'll have few of the code snippets already available.
You clone that, create the copy and then we will use it, right? And we will not do this for all the things, we'll only keep it for some of them. OK, so can anyone tell me some of the use cases that you know that you think is NLP based?
Apart from chatbot and virtual assistants.
It can be anything.
Your language.
Translation where you have one language conversion to a different language. That was actually the use case used to build Transformers, right? So if anyone know there is a paper called Attention is all you need, right? So the purpose of attention, all you need is to build a neural translation and test it.
How well it is you have encoded, you have decoded. So any other use cases?
Sentiment analysis is one of them. We will also build one of the application today. So what does sentiment analysis do? You have certain reviews of product and you just have to tell whether it is.
Uh, positively reviewed, negatively reviewed, and if there is any emotions behind it, that has to be displayed. So one last example we'll take.
Someone else. Someone else.
So 2 use cases are done, one more use case.
Thanks.
NLP based pure NLP that will involve OCR also.

Pure NLP use case like for example you have chatbot right? Uh, next.
Text auto completion. Text auto completion. Usually it's like next word predictor, right? What is next word predictor? You we go to Google, you search for something and now you are getting what can be the possible next word. So that is auto completion. So in short it is next word predictor.
And this is something you call as autoregressive models. Next word predictor is NLP. So this kind of models are called autoregressive. GPT is autoregressive. So what does GPT do? It generates 3 to 4 tokens and then it predicts what can be the possible next token.
All right, so the main focus for today will be this two concept. One is tokenization and one more is normalization. We'll take an example of what use case you only gave, right? Sentiment analysis. So let's suppose I have a.
Review this review is this product is great and some blah blah blah. When you write this entire review there are few keywords right? So the keywords might not be relevant like this is right? So what matters the most?
The words like, hey, this is a great, great is a a keyword which matters, right? Let's take one actual example, Flipkart.
Review on.
So you have awesome laptop. It's super fast. The build quality is remarkable, one of the finest product of Apple. So now if you see it's then the is one of the. Do you think this keywords matters when you are building a model?
So I'll give you one one more case study, right? When you're building a model, you will obviously require lesser number of tokens to give so that you can train a model. And most of the time you'll have paragraphs of reviews, right? People will write a bunch of paragraphs.
So some people will just write 2 words. So you'll have to manage how much tokens you're passing and you have to standardize the length of it, right? And standardizing length can be removing common keywords. What if I give a URL in this?
Let's suppose I give a feedback and then I have a URL. Do you think model can look into that URL, check and then predict for sentiment analysis? Will UR matter URL? For search application it will matter, but for sentiment analysis no.
And third thing, most of the time you'll have upper case, you'll have lower case, right? Initially, whenever you build models, this upper word and lower word will have two different tokens, right? So this causes an issue. So you are converting your entire text into lower case.

So converting your text into lowercase, removing stop words, removing the extra access of URLs, all this comes in normalization which is used for text pre processing, right? And it can also be UTF 8 encoding, right? Some of the keywords might not be relevant.
So you have to encode it first. So that is also one of the normalization techniques. We also have one more technique which we will cover today, which is stemming and lemmatization. I have few slides for that. I'll come to that shortly.
With this basic NLP.
Like even if you're removing the word like I the to change the context. No. So for prediction it won't change right? So if you're you can write a review. I am awesome. It is not about the product, but then it will say it is a positive review. Yes, positive review.
So which is not? I can give it one star. I can say I'm awesome. This is ****. What would it say? So now this is different when it comes to the entire sequence based on whatever trained model you have seen. Let's suppose here you're not checking the starring, right? Star is a different logic. Here it is.
Only based on text. So if the sequence has this is awesome, this is ****, probably it will score it as positive as per it goes left to right sequence right? So it won't check what is the second keyword. It will directly give you it's positive.
And probably since **** is also mentioned, you will have lesser positive you have confidence score. The confidence score will go down.
Right. So every single review that you add will have a confidence score and you can define a threshold. If the positive score is less than 75%, you can look at a manual review of those, right? If it is more than 90%, you think OK it it is legit.
Right. And same case in negative also in negative, if it is 60% and 60% negative means there might be some kind of issues. So you look into that review manually. If it is 90% negative, that means it's negative.
So that confidence score is manually reviewed sometimes when you have the logging dashboard, so confidence score will come. So these are some of the normalization technique which you use before you train the model. So basically when you train the model there is a rule called.
Of dimensionality. So curse of dimensionality in the sense whenever you build a model have lesser features, right? In NLP, what is the feature? It's text. So let's take second example.
Everyone knows what is Kaggle.

So let's suppose I don't have text, I have tabular format, I have movie name, I have rating, I have genre, then I have runtime and I have other details. So now the task is to predict whether this given movie.
Will what you call will do good in market or not, right? So what is the prediction here to predict zero or one? If it is zero, it didn't do good. If it is one, this movie will do good, right? So this is just one example.
In order to predict this, do you think you will take all the features that you have? So here the features are rank, name, year, rating, genre, certificate, runtime, tagline, budget, box office.
So do you think rank will matter? Rank in the sense here it's just serial numbers. It's index number. Do you think you'll pass serial numbers to a model? No. So you're reducing 1 dimension, whereas in text every single keyword is a dimension.
So now you understood the difference. So in order to avoid curse of dimensionality, you need to reduce your features. So this features in text is a keyword and that is mainly regarded as vocabulary.
Which is a single token. Is this understood? Yes. So dimensions in text, which is NLP is mainly regarded as vocabulary. So you define a vocabulary size. Let's suppose I'm building sentiment analysis predictor.
How much vocabulary can I give? I'll give 10,000. Now the model that I build will only have the vocabulary of 10,000. If he sees a new word that will be marked as unknown.
So that 10,000 is the limit of the model. Hey, I only know 10,000 words. I don't know more than 10,000. So even if it sees a sentence which is completely new, it won't do well, right? That's the reason why many people start improving their vocabulary size.
So this is mainly used for how much tokens can one utilize. So normalization vocabulary. Now coming to tokenization, I have a sentence. So this is a sentence. This entire sentence will be converted.
Preprocessed and once preprocessed, you need to convert it in a single token. So let's suppose I remove all the stop words.
Awesome is 1 keyword, which is 1 token. Laptop is 1 token. Super fast is 1 token. Build quality, remarkable are individual tokens, right? So let's talk about tokenization because it's very interesting topic.
So when it comes to tokenization, every one of us needs to know what is difference between a word and a token, right? So every one of you are using LLM's API. So if you look at Open AI.

What is the pricing of open AI? On what basis that pricing it tokens 1,000,000. So do you think 1,000,000 token equals to 1,000,000 words? No. So tokens is different, words is different and if you want to know what is the difference between that? You can just use this formula. One token equals to three by 4th word. That means if you have 75 words in a paragraph, that will be equivalent to a 1000 tokens. So this is just rough estimate. It's not that accurate.

So let's take one example here. How many tokens do you think is there? And you have to tell me what is those tokens. So no open AI tokenization, open AI tokenization load. I'll repeat again.

The purpose of using the sentence here is just to tell you whenever you see pricing and if you encounter a keyword called, we support 1,000,000 tokens. 1,000,000 token is not equal to 1,000,000 words and if you want to know what is the rough estimates. You can just do this formula right, which is 1 token equals to three by four word or vice versa, which is one word equals to 4 by 3.

4 by 3 token. So now here you have one example you have to tell me.

How many tokens are there in this sentence? So how many words do we have? 333 words. Now how many tokens?

Don't do. Don't use the form. How many of you are saying 3 first 1-2? How many of you are saying 412? What about you? 4 is 2? OK.

Five, five, five. OK, so why did you say 3? You went this one open, one open one. OK, so why did you say no? So I I you just roughly estimated that. OK, because of the formula. OK.

But what if you see what you said to 1 token? In your case, if you said more characters, there are three words, right? Words will always be smaller compared to tokens. Tokens will always be more. So the question here was how many tokens is there in the sentence?

So there are three words. You can never have tokens less than numbers. So here if you see the numbers of token is small. So now we want to change your response. One, one, one, one. How one token can never be lesser than.

OK, so or if you say 3, that's valid enough. If you said two, that's not valid, right? If you say 4 again, it's valid. If you say 5, that's valid. If you say 6, that's valid. OK, OK, more than three is valid. More than three is valid, right? But how many?

You can use that formula that is only for estimates. So now can anyone tell me how many tokens? 5? Why 5 open one AI2 token 3 organization 4 and lower five OK.

It's six to seven. If we go by 4 characters approximately, then it's seven. OK, so you're

saying six, right? Usually it is 6, right? But.
As per formula, never go it right. As per formula, it's only when you're doing pricing calculation. OK, but for this example, if it is sentence to sentence, if you want to see, we can directly make some assumption, right? Six is a better assumption, 6 or five. But this one will be 6, so there is a tool layer.
Which has open AI tokenizer.
We can take up nothing, nothing. You send your lines with the same program tokenization, then organization. More people and grams. Makes one.
That's a different thing. How does tokenization work over here? So here I'll explain the algorithm as well on how this algorithm works. So there are three tokenization techniques.
One is Treebank which we will look today. So what Treebank will do is once it sees a space that is 1 token. So as per Treebank the number of words is equal to number of tokens, but if you are using.
If you're using any punctuation, punctuation will be considered as a token, right? Open AI, they build their own library which is tokenizer which is called as BPE which is byte pair encoding.
And you also have something called as word piece, right? So you have three bin which is pretty old, no one uses it. Then you have BP which most of the people are using it now, and then you have word piece.
Uh, there is a YouTuber.
OK, not me. Since I told YouTube, it came my name.
So I'll just open this guy's profile as well.
So if you look, I'm not promoting this guy. I just want to relate the topic that I'm learning. So he started from Stanford, then worked at Open AI, then joined Tesla and then worked at Open AI again. So he has a YouTube channel and his YouTube channel he shared one video.
Uh, this one.
So let's build the GPD tokenizer where he actually explained how BPE works in detail, right? So we will take one smaller part of this and we will build it tomorrow. Today we will use Treebank. Tomorrow we will just, I mean, we'll not build tokenizers from scratch, we built it from scratch, right? We will just use the existing database and use a core library.
Knowledge, right? So obviously there will be difference here. This guy actually coded, brought the data set, trained it for longer time and build that tokenizer. He does this

for all his videos, right? I didn't want to promote, but this is something that goes above you already if you don't know Pytorch.
So Pytorch is 1 library. Once you learn it, you can learn how to build GPT from scratch and then this tokenizers. You also have some what you call non tech related videos where it doesn't code like how I use LLMS.
And if you look at his use, it's like crazy. And as I said, most of this goes above your head. If you look at these videos, it's like in 6.1 million like who will sit and code it. So once he says anything, right, who do you think vipe coding gave the name?
So if you search for vibe coding, it will tell this guy give that name. So if he says anything that's done, you just have to believe it. It's like that particular person, right? So if I just search for vibe coding.
Oh, so if you see it comes, it's now all good. So while coding, I'll do Wikipedia.
So if you look at the first line, why coding is an artificial intelligence assisted popularized by Andrew Karpathy. So if he tweets anything, people just believes him blindly. So it's like that personality. So the reason why I want to.
Wanted to show this is tokenization has different ways you can do it. One is Treebank, then you have VP and then you have word piece. If you want to learn word piece in detail and VP, I'll share the algorithm on how it's built. But if you actually want to code by scratch then it's that video.
Hold it. So coming back here, if you see you have open AI, open is 1 token, AI is 1 token, token is 1 token, ization is another one. Then you have lower which is one more and then you have deep. So there is no.
It's a token. Oh, it's highlighted in purple color. OK, so if I make it as Ed.
It's fine. So there is no you can't predict like if there is a word how much tokens it will be there. This was just one tricky examples which I took where I could have shown the difference between words and token. Now how this works? It depends on the data it was trained on.
So if other data was used in which Open AI was considered as a single word, you could have got Open AI as a single token.
So this depends on the data that you train, right? Tokenization also needs a data and once you train that you'll have, how do we split it? Is this clear what tokenization does?
No, you'll have to train it. No, no. So how will I know what?
No, it's not a dictionary. So you have billions of data. Once you have the data only, you can build these models. So we have BP tomorrow. In BP we need an actual data

set to train that BP. So what you see here.
And what you see in our VP example might be different because we don't have that much data to train it.
Yeah. So if I am working in an at a financial institution, let's say, and we want to train it particularly for financial data, so it makes sense to train our tokenizer.
For financial data, correct.
You will have to tell him. I'll show one example rule. So usually let's suppose we built Luxembourgish LLM, right? So if you want to build something, let's suppose Luxembourgish.
Every LLMS has some level understanding of in the Canada.
Famous languages which has publicly available data. Luxembourg, Luxembourgish in their own country. They don't speak Luxembourg, they speak French, they speak German. So for us to build Luxembourgish language, we had to build our own tokenizers. We have to train it. So you have a data.
You train the tokenizer and then you go to the model, right? So when we learn hugging phase, we will use Mistel model. If you want to use Mistel model, you need to have Mistel's tokenizer. If you use Gemma's model, you need to have Gemma's tokenizer. So you can't use Mistel's tokenizer and then use a.
Gemma model you will get unwanted errors. So if you're using any model, you need to use their own tokenizer and then no model.
I not using GPT price tokenizer instead I'm using CKE 10 oh from bit tick token. Yeah tick token is different. Tick token is again maintained by open AI. OK right. So that API based they're managing it from their end. But I'm telling if you're using something from open source side you will have to.
Think from your end itself. If you're using hugging face, they have auto tokenizers. So auto tokenizers will take the heavy lifting on their end. So that is framework to framework.
Huh. If you use separately, so there are two different classes here. One is Gemma's tokenizer, one more is auto tokenizers. So if you define Gemma 3 and if you define auto, you're safe. But if you define Gemma's tokenizer and then if you define Gemma 3.
Then fine. But if you define Mistral where you have already defined Gemma tokenizer, it's an error. This will make sense when we show a gameface example, but in short tokenizer can be trained.
When you need better performance. So let's say if I'm using Gujaratis, we if we use

UVFA, we don't have Gujarati, but if we are using Unicode then we have Gujarati now not much.
Opens will be for Gujarat, right? So we need to end so it does not end up as an error. Is there any reason? Because it might not have been trained on Gujarati. Oh, can you repeat?
Let's say we are using who is my friend like model those that is some sentence and I'm passing it to tokenizer which is not OK.
So it will have more tokens. It will use more tokens because it does not understand. I think no. So if you are passing this sentence, it won't be an error. It won't be an error. It will. So if you give a sentence, yeah, which is my friend is something, yeah, it will convert that into tokenizer. There might be high chances that tokenizer might not be valid because it didn't see that word. So now that word will marked as.
Unknown. OK, so there you are using vocabulary. So vocabulary in the sense every token is a vocabulary. Yeah, yeah. If it has not seen that word, it's unknown. OK, so it will just mark as one token. Unknown is A1 token. OK, padding is 1 token. That's something so.
The more set of characters you can edit, the more set of characters you can identify, it will be a different token. It will become character wise token. So if you open that there is more token, there will be more different for the token.
Can I?
And just like anything, anything, anything without space. Yeah, no, without space. And now if you see the number of tokens have increased because it is trying to collect the set of word with missins and that is a token. So if you see VD is 1 token, DN is 1 token, VK is 1 token. Probably it might have seen those.
Words in the data it was trained right? So here I wanted to show one example. Tokenizer dot Jason.
OK, so I'll just tell you one thing. So as soon as you log in here, right? Oh wait, let me do that.
I was considering it as a mix. Then that's why I mean I will not show you the thing in person like it would be a big no no no. So somewhere and very right. So it doesn't so usually you.
You have tokenizer. You have model. Tokenizer can be trained. We train it for many. Tokenizer is a model. So BP as I said word piece, not one piece, but word piece.
Where do I have said my password?
Uh, and then it was open AI to embedded embedded embedding.

OK, let's look at this example and then we'll continue.
So can you see this Lux Lama?
So for Lux Lama, as I said, there is no publicly, I mean most of the model doesn't even understand Luxembourg. So we took newspaper, we took magazines, we took most of the time it was newspaper itself. We took those newspaper, we took those articles and we trained tokenizer.
And once we train tokenizer with the trained model, we use the model as well, right? So model again is trained. So that's how it works. And now coming to the tokens, right?
If I open Gemma 3.
And if I go to tokenizer dot Jason file.
So if you see you have something called as pad. Now I'll give you an example. There is a review which has a one sentence. Then there is one more review that says how may seem good. Now how do you build your dimension? Dimension needs to be same.
So if there is any two word and one more as 20 words and one more as 80 words, now we are building a dimension which has to be only 50. So if it is 2, your remaining entire thing will be pad which is 000. If not you can use this. So if you see what is pad.
So it will fill with that particular value and if it is 40, remaining 10 will be pad. If it is 80, pad. So if it is 80, it will truncate. So from where you have to truncate, you can define that truncate from right side or left side if you said left side.
As a you don't have to give this. This is defined in a tokenizer. When you fine tune your model where you have to tokenize from left side or right side, sorry not tokenize pad, right. You have to define whether you want to pad from the left side or the right side.
So if you tell left side 000 last two will be your actual tokens. Now if you tell a key then and if you tell pad to be left, you're starting 30 will be removed. So you understood the importance of pad. Just like pad you have us.
So why do you think open source elements sometimes you garbage results? You don't define end of you know prompt pattern. Have you always seen prompt template for Gemini Cloud Open AI? You never use prompt templates.
To directly give you a prompt and you're getting results, but if you're using open source, you have prompt templates.
Oh.

Gemma 3 Nicole.
Your birth.
So one second.
Wait, what happened? Because you connected the project. Oh, sorry.
But this is something we'll repeat in open source elements because we have one session for that. So here if you notice, if you don't define this startup and VOS end of end of term, you'll get an error. Not error, but let's suppose you ask what is the capital of India?
It will tell you to release capital of India, then again some additional tokens. There is no end to it, right? It's because we never define a problem. Every single open source LM needs to have this if you don't define it.
There might be some inconsistency results. So here if you see BOS which is start of sentence, you have end of sentence. So these are some tokens which you when you're training your model, you train this right padding end of you can give this your as well.
Chat is not closer, so we don't know what token they have used. OK, so some of them just use E like Mishel is there. Mishel just use INS.
Now Gemma used VOS, so we don't know what open AI has used because it's closed source. We can know of Gemma and Mistel because they're open. We can check their configuration file. This doesn't apply for Cloud, Gemini and Open AI, right? So.
Padding, VOS, EOS and we have unknown. So if there is any token which is not in vocabulary, it will mark it as unknown which is UNK.
So if I come here UNK, so this is added as UNK.
You have mask. There are multiple. All these things are there, right? This is called a special tokens.
EOSBOS which is beginning and then UNUNK is for unknown. Is this clear?
OK, we will obviously look at those examples. We have open AI, I mean how to use ugging face. During that time we have how you can use auto tokenizers and then model. You had any question or it was related to screenshot, OK.
And what are the libraries that we can use to tokenize? This one is the most famous which is NLTK which we will be using and in some capacity we'll also be using Agging phase and the other libraries are fixed blog, Gensim, ACE, Keras and Pytox.
So these two libraries are very heavyweight in the sense there are too much of dependency, right? And we have to come when you are going to install this. So this can be only used when your entire pipeline has models which is running on GPU.

If you're just working on lightweight apps, then you can go with energy, OK?
Text BLOB and Gensim. They are there, but it's not that great libraries. The performance is not great, but these are libraries where you can still use for experimentation. But I will not recommend NLTK hugging phase are the best choices. Because Hugging Face is something that is maintained daily, right? They have Transformers library and they have a very solid team. So that's the reason why Hugging Face NLTK is a popular library of Python which is existing probably since it has started. So NLTK is there.
Now what we will do is let's understand some of the core concepts of NLP before we can jump into NLTK. So now the focus is we know the library to choose and what NLTK provides. We will just understand that functionality.
So this is something everyone of us know. One is POS which is parts of speech and then we have NER named entity recognition. So NER is nothing but we play the right name plus animal thing. It's related to that.
So now what we will do is let's just revisit our grammar so that when we write code we will have some understanding. OK, what is happening? So I didn't wanted to pick POS, but in order to understand POS tagging, you need to know parts of speech, right? Just some part of it.
You have the sentence Harry Potter studies at Togwarts. What is the parts of speech of studies?
Verb. So what is verb? It denotes what the noun is doing. What is the noun here? Harry Potter. Harry Potter. So it's verb. So what is the parts of speech of genius?
No, no. For this sentence it is known for this verb. But if you look at at this sentence, now look at the sentence and tell.
Yeah.
It is adjective.
Now this one like works extremely efficiently, adjective, adjective, efficiently, it's about no what is the what is the efficiently is.
Adjective and adverb. Adverb and adverb. Sorry, both are adverb. It adds to the verb. So and then again it adds to the verb. Yes, I'm very bad with grammar. What's the verbs and what's how so?
And extremely how so that's why add work adds. So now what we will do is we understood POS, but POS is just small part of POS tagging, right? So whenever it comes to NLP you have two.
Core concept one is schematical understanding and one more is morphical analysis.

So schematic is mainly dependent on how you tag the particular words. So you create a tree like structure. So the tree like structure when you create every word is tagged.
And then you have morphological analysis, which mainly depends on grammar. So these are two things that is mainly used when you build a model, right? So now what we will do is every single sentence that you have. So let's suppose you have input. Harry Potter studies at topwards. Harry Potter studies at topwards. It's X1X2 till XN. Now every single sentence that I have, it's have its own, like Harry Potter is a noun. Now in noun also we have different noun, right? You have proper noun, you have common noun. So you define that.
And for every single word you have byte. So let's just take a few more examples. But before that, can you take this image? Just take the photo of this.
So either it can be screenshot in your laptop or phone and as soon as you take screenshot just keep this screen active in your laptop.
If not this one, what we can do is I'll show one more.
Photo.
You can take the photo of this one, which is much better.
Keep this one active.
So I'll show one example. Remaining example you'll have to tell by looking at that photo. So which is that photo? It's this one, right?
Now I'll just tell what this photo is is doing. So if you look at if you encounter noun, you have different noun. One is common noun which is NN. Then you have NNP which is mainly used for proper noun. Then you have NNS which is for plural.
And if you encounter what you call adverb, so if you notice adverb it should be RB somewhere RB. So can you see RB is for adverb and if you see adjective it is JJ and for verb you have two things, one is.
RBZ which is for plural and then you have RB which is a normal base. So if you see here eat it is VB. If you see eat it is VBZ. So now you've got. I'll show some sentence you just have to tell me.
What is the tag of it?
For example, here I have a small dog barked loudly, right? So what is a? It is DT which is determiner. Then you have small. Small is an adjective. Then you have dog which is NN. Then you have barked.
If it was bark, it would have been VB, but it is bark. It is VBD. Loudly is an adverb which is RB right? So you can just cross check in that photo. Now can you tell me for

Harry Potter loves awards?
Yeah.
And then it's same one now.
Laos is VB, VB, then Chris, it is Laos.
Hogwarts is what kind of known?
It's NSNMB and then.
OK.
It's proper non only. So you have Harry Potter, which is NNP, Potter is NNP, Laos is VBZ, right? And then you have Hogwarts. Let's take one more.
Let's take two more to the story yesterday.
Yeah, he's a proper now, so personal and then and then B. It is singular, so she's NNB. Quick please.
RUE OK PRB. Yeah. PR and then quickly is RBWB CRB. Ran is what? W W No, no, not Ran.
VB C VBD for VBD for random. OK, two VBD is fast tense now. VBD who is it too? So can you also see Ian there? Ran is VBD. Can you see Ian in your? What does Ian say? In by So So what? Where is Ian? OK, two is 2. Probably you also have two, huh?
The the is the the store is.
No and then add.
And yesterday.
Store is and then and then and yesterday. Yesterday also no in this sequence probably it will be at work.
Yesterday.
What is yesterday's?
Yesterday is a noun. Part of speech is a noun, a noun and an adverb. So in this sentence I felt it's an adverb.
Yesterday I think it's an. So you have PRP which is proper this thing and then quickly which is RB ran is VBD which was correct. Two is 2, there is DD store is and then yesterday is RB and then we have dot which is function.
So if this here.
Uh, where is PUNC?
Do we have UNC?
OK, in this it's not there, but you can ignore punctuation as well.
OK, I've added two more. OK, this is last.
Yeah, I see the new software JJ, JJ, that is very simple.

OK, this is just to bring out of your sleep before we can go to code.
Only parts of speech is wiki named entity recognition. It is like child spray.
For named entity recognition, I'll ask someone who has not spoken yet, so this is our last chance to speak.
No, no, she she is. She is active. Uh.
You three and four.
Yeah.
OK, DT is not valid. OK, let's say for work done perfectly. We'll proceed next.
Perfectly. Perfectly is adverb. So you start with adverb. The everyone said new is adjective, software is a noun verb. Again, it's a verb. And then you have adverb.
Is this clear?
New is adjective. It is saying something about the what kind of software, new software. So just to summarize what we did. So usually most of the NLP concept that you see depends on lexical analysis.
Then you have which is mainly using schematic. Now these two keywords are very important when we work with RAG.
So lexical and one more is schematic, right? So if you see something called a keyword search which is using keyword to keyword, you will encounter lexical as a keyword. Schemantic is mainly vector search, right? So now probably it might seem OK why are we even learning this? But when it comes to high?
When you build you will see this concept lexical and schematic right? Now coming to the another scenario I said morphical, right? Morphical analysis is just to is this sentence even making sense right? So for that you have morphical analysis which tells model or NLP based model that is OK.
OK, whatever text you're passing, does it follow the rules of modifical analysis? That is to check the grammar, right? So now POS tagging is done. For every X1 word you have Y1, which is the tag of it. So the tags are the 36 tags that you have seen in that image.
For same you also have named entity recognition which only falls under 4 category. One is person, one more is organization and then you have location and then you have geopolitical entities. So you also have examples for this. So if you see turning is a giant of computer science, this is a person.
Person and then you have the IPCC warned about the cyclone. IPCC is an organization. Then you have mountain centers is in Sunshine Canyon. So these are location and then you have Palo Alto is raising the fees for parking.

This is a geopolitical entity, so you will find NER in just four. One is person, organization, location and GP. These two things are very tricky. Sometimes you will this this will become very tricky.
If you encounter Tokyo, you might not know, OK, this is a location or a GP, right? So whenever you see location and GP and if you think, OK, this could be reverse, that's fine, right? So usually even I'm not sure why they keep this as a separate thing.
You just think LOS, LOC and GP are a location, right? So if you just mispronounce it or misconfuse it, which is completely fine. So here we have an example. Naruto trained at Konas Ninja Academy. So Naruto is a.
PR person and then you have Konoha's Ninja Academy as a organization. So now you have this thing Goku for Vegeta in West City.
No, the city location. It's not a country or zero location, right?
OK, wait here. I'll have to create a copy, but this is something which is easy, so I'll just show it. So we have the Phantom Troops is feared across the world. So Phantom Troop is a organization.
Now if you look at all these examples, you had multiple words also. Here we just looked at Goku and Vegeta fought. What about fought? What about in right? So you have one more sub rule in NER tagging which is.
BIO for example the phantom troop. What is the starting of organization that is labeled as beginning B org and what is inside it is tagged as I org OK phantom troop. Phantom is Biorg, Troop is Iorg. Same goes for names. Harry Potter. What is the beginning person and what is the inside PR? And if you see something which is not a named entity recognition, it will be marked as 0.
Which is outside any entity.
Is this clear?
So we'll take an example so that it's clear. So here you have an example that says Sundar Pichai is the CEO of Google and studied at Stanford University in California. So what is Sundar? It is the beginning of a person which is a named entity recognition.
As per previous example, what could have you done? You could have said Sundar Pichai person, Google is organization, Stanford University is a organization, California is a GP. So this is how you could have said.
Now what you need to do is you need to break it down.
B is beginning, I is inside, O is outside entity. Now this example makes clear, so you have to tell it for this one. V per, I per.

What is joined? OO joined is O.
No, Intermiam is a organization.
I org, I, B org and I org I org after oh oh oh.
Paris might come GP or might come lock. We are not sure, but both are valid response.
Uh, yeah, Saint Germain. It will be a single, yeah.
Why? Because it's a single token. Oh, oh, oh, oh, the same Japanese word. No, it will be.
Why BI? Sorry. Yeah. So if you see Paris will be B, this will be I I I.
Yeah, club this is OO club is over. If you say Barcelona club, probably it will be inside, but still Barcelona will be tagged as one. Club will not be counted.
Club won't be counted. Club based France. France is a location. France is a location here. It should be GP, right? GP, yes, definitely.
Oh, club is only. So you're saying club will be organization?
OK, you said. Yeah, I said. Oh, even we said.
We meant Barcelona Club. I thought you said organization, but is this clear? Intermia means organization, then Paris, Paris and German is also organization. OK.
Is it clear? We saw two things here. One is POS tagging, which mainly depends on 36 tags that you saw, and then you have NER. NER stands for named entity recognition. We have only four tags, one is 36 and one more is 4.
Now this is the last topic before we go to code. This is stemming and lemmatization. Before I spoke about converting your feedback into lowercase, removing extra URLs and removing the stop words which are commonly words.
Then there is also one more preprocessing technique that is called stemming and lemmatization. So let me give you one example.
Running runs and what is the superlative ran ran right? All three words. What is the stem root word of it? Run right? So if you bring all these keywords into a single. Word. You are reducing the dimensions, so there are three vocabulary running, runs, ran, and if I just define that as run, it's still the same meaning, right? So that is done using stemming and lemmatization, but both are different.
Stemming will ignore grammar, right? It will bring it into root word. Whether that root word makes sense or not, that doesn't matter. That's what stemming does. If you want the root word to make grammatically correct, that is a lemmatization.
The goal of both stemming and lemmatization is you have a word, get the root word of it, right? So there are some examples you have running.

Whatever you see on the first column is for stemming.
This is stemming and this is lemmatization. So for running if you see both are doing correct then better for stemming it is still better but lemmatization since it follows grammar. Now what is that keyword I said I told 3 keywords lexical.
Schemantical and one more.
One more keyword I said which starts from M.
Morphical analysis. Thank you. You opened your voice. So and then what do you say then will fall under morphical, morphical logical analysis, right? But why? I didn't understand that. Which one? So when it comes to studying, there is no rules that it follows, right? You are also explained on Wordnet data set.
So there is a bit of what you understand.
I mean, you understood. Yeah. So if you see for flying, just say like flight, which is wrong. OK, so there are better to boot when it got better to boot. Yeah, so most of the time it will skip some keywords. So it just thinks that OK, this is the root word of it.
The root word of better for that is good. Good. Huh. OK, so this whatever we have here, sometimes you might not get the right what you call right root word, right? It's alucinets. So whatever you see here, there is a popular data set called Wordnet. Word net. So based on that this particular model was stringed. So during that time probably those keywords were missed out stemming both stemming and lemmatization. So these are not pure what you call.
Pure algorithms, right? So when you build NLTK for NLTK, you'll have to install that particular library which is NLTK download word net. So once you download that word net then only you can utilize these examples. Is this clear? So every single logic algorithm that we will implement now.
We have to install the framework that is install the package. If it is stop words, let's suppose you want to remove stop words. So from where are you getting the data set? So for the data set you have to download that particular data set. Same goes for Wordnet.
Is this clear? So you have was. So if you look at was here it is just WA, it is B right? So this is how stemming and lemmatization works. The core logic is get the stem word. Is this clear? This is used for text reprocessing.
Converting your data into lower case, then removing hyperlinks and what was the next thing? Stop words, stop words and then stemming and lemmatization. So it does both stemming and lemmatization. So we usually don't use stemming, we use

lemmatization.
So stemming is usually not used in most of the frameworks, but stemming exists. Stemming is just to show you the difference between what stemming is and what is lemmatization.
Because this usually doesn't make grammatically correct.
So I'll just share this particular notebook.
So don't start with Numpy. Numpy probably will cover in the online session because I thought Numpy we can do and then we'll go NLTK. But since you are starting with NLP, we will focus on NLP and whatever is non NLP related we'll cover it in online, right?
Did you share it? I'll share it in here.
So let's create the copy of this and then you have something called as NLTK.
We'll go step by step and as I said, whenever it comes to NLTK, every single logic that you see, whether it is POS tagging, NER. So what is NER? Named entity recognition, POS tagging. How do I know this POS tagging is correct? This NER is correct.
So every single logic that you have, whether it is POS, NER, stemming, lemmatization, they have their own data set. You have to.
Like uh PIP install NLTK. Oh for collab it's not required, but if you are running it on VS code you will have to do PIP install NLTK.
So as I mentioned earlier, you have a data set called Wordnet. This is a common data set that was used for POS, then NER stemming and also lemmatization. So this you have to use as soon as you import NLTK.
You have to use Wordnet download. It will download that particular package.
Uh, like this there are many more which we will we'll be covering down as well, OK.
So if you see something called as lookup assertion error right in NLTK and if it tells you to download any framework, that's not an error message. It's just that you have to download certain package before you can use that particular algorithm. And in one of the algorithm I purposely removed one download statement so that you can see.
That error error message.
So are we on the same line? OK, so if you're using VS code, you'll have to use PIP install.
NLTK and here you can just add Wordnet is.
Mainly one of the data set.

In the NLTK package.
So everyone created the copy of this notebook, right?
Uh, environment Python 3 iPhone M.
VENV which is virtual environment space any name. So the common names are PPNV or just ENV.
Is it done? What happened?
So these two commands is the right. These two command like let's suppose if you're using VS code, let's have one file as requirements dot PY, not requirements dot TXT. Either you can keep it download dot PY or requirements dot PY. All these download statements can be in a different file.
Right. And once you download it then you can have a main dot PY. So it's better to keep it both separately because if both are in the same file, every time you run that file it will download that model. So this should be just one time process.
This should be just one time process.
Now here there are different types of tokenization as I said. So initially we started off with the word tokenization that every word is equivalent to.
Token and then you also have sentence tokenization. So what sentence tokenization will do is as soon as it sees dot it will create a different token. So one is word tokenization and one more is sentence tokenization.
And you have a simple sentence here, which is NLTK is a powerful library for national language processing in Python, right? So how many tokens does it have here?
Ease and everything or just the dopest that we considered like 123-4567, three for a six. No, now we'll not have to do preprocessing right since we're not building any application. So here the goal is you have.
A sentence you just have to tell how much tokens it is utilizing.
1234567891011 No you also have dot here.
So it will be 12, but let's just see how BP does, whether it has increased number or lower.
You have 13. It has made NL as one and TK as one. These is 1 token A powerful library. So usually it's 12, but here it's 13 because it's using BPE right? So as per VP logic NLTK is a two different token.
So if I come back to the code, NLTK is a powerful library if I add load here.
Here now it is 13.
Here it is 15. Why? Because this one word added two additional tokens.
OK, so how much sentence token is does it have? What's the length of sentence

tokenize?
Sentence token. Sentence. Just one.
So from NLTK this syntax is dot tokenize and then you have import word tokenize comma send tokenize.
Sent sent is nothing but sentence and this is word token is.
Now if I print sentence instead of print, if I do length of sentence it should be 1.
Is this clear? Sentence tokenize. Now for word again I'm doing tokens equal to word under score tokenize sample under score text which is the input that I have. You have NLTK is a powerful.
After every single space it is considering every single word as a token. The only additional thing is if you encounter punctuation that is added as a token. So here I'll use comma.
Space.
OK, even if I don't give space, it is at open.
So if you see NLTK is 1 token, Ease is 1 token, YA powerful library load, Kama is a new token. So now how many tokens do I have? 123456 Kama is 7.
8910111213 and 14 Why? Because comma and dot are also considered as tokens. So instead of print, I'll just add length. The sentence is full stop.
Push up.
You probably you might not have added load, load, load and comma. So these are two words that we added, OK.
And this type of tokenization is called Treebank, right? So Treebank was the initial tokenization algorithm that was built. So even if you look at this diagram right, you might have seen this keyword.
Pen tree bank core 36 parts of speech tag, right? So this pen tree bank is the algorithm.
So now what we are trying to do is we are defining tokenize then treebank. From treebank I want to use treebank word tokenizer and if you want to detokenize it again you have this detokenizer but this will be similar to your input token right?
Tokenize Treebank.
So treebank is an algorithm name. So if I remove this and if I tree dot.
How about open ice?
You also have something from Stanford.
So we will be using two more tokenizers, one is VP and one more is word piece tomorrow.

So here we are using the same example which is open AI lowered tokenization lowered so and then dot how many tokens do I have here 4?
Open AI tokenization, load and dot. How many tokens will BP consider?
1/2.
3456 and seven total 7 tokens. Last time when we checked this example, we didn't give full stop.
So the syntax is same. You first instantiate it like word tokenize and once you instantiate it, just use a function called tokenize.
So this syntax is imported. Once you import then instantiate it. Once you instantiate you have a object. This object has a method called tokenize which takes the input variable.
What happened?
OK, so again, if you do D tokenize, it's just similar to whatever input variable you pass. Still here. Anyone has any doubt? We have word tokenize.
Which is similar to the treebank algorithm and then you have sentence tokenize which will split based on sentence right and.
Nothing, I just there is one function for it.
So as I said, D token is will give you the your same input variable. So if you do double equals to S, it's the true.
Oh.
No, this is for if you give BP right? If you give BP then it will make sense. For BP detokenize makes sense, but for what do you call?
Treebank. It doesn't make that much of the difference, but can you arrange construct back? But that construct back is your same input variable.
So if you give this line, it should be true.
Probably we have to battle test this as well, like how well or true this holds.
Because when I came to load, I didn't knew that load has two tokens, right? Because it doesn't even make sense to have this word has two tokens because D just one alphabet as a token doesn't make sense. So you can just imagine whenever you use open your API key.
Lowered is 1 token just like this. How many tokens are there which is going waste right? So using tokens you have to be very cautious.
And like this, if you have around, let's suppose you are, you are sending one paragraph which has 20,000 tokens out of those 20, 20,000, if you have 10,000 like this, it's just waste of tokens, right? Even though it's a small amount, it's a massive

amount number.
Use this bad grammar. Then they'll pay. They'll have to pay, but still 1,000,000 is a big count. For 1,000,000 how much you're paying? $0.00 is the price of GPT 4.
4 of mini is $2.00, Sorry, GPT is no, it's different.
So this will also be counted in input tokens and as well as output tokens. So this is the logic that they use.
OK, now we'll go with POS. In order to define POS again, you have average perceptron.
So this thing, right? Let's suppose you don't know what to install. Don't give this line. I will just delete this and now what I will do is I will just import this statement.
From NLTK import POS tag. Now what is this? It's called parts of speech tag.
This is similar to what you see in your image, the 36 tags, right?
So here you have a sentence.
So how did we approach? How did we approach to POS tagging? I'll just open that image. So for POS tagging, the first thing is every single input variables is converted into tokens if you see X1X2XN, right? So if this sentence is a small dog barked loudly. You need to convert this particular variable into a token. Once you convert that into token, then you have to map, right? So what is the logic you need to use now? Let's suppose this is S.
What is the next line?
Word tokenize. So you have to use tokens equals to word tokenize.
S and then.
Your POS tag off.
Opens.
Is this clear? So I'll just undo this.
So first thing is you have a sentence. For sentence you have to word tokenize it. What is the length of tokens?
12345678 and 9:00.
It's 9 right? Now that you have tokens, what you need to do, you need to pass that tokens as a input variable for POS tag.
This should give me an error.
Now what is that error name? Look up error, right? Whenever you see look up error, that's not an actual error. It will just tell you what to download, right? So you just have to copy this. You have to create a new file. I mean whatever download.py is there, right? Comment your previous download and then just run this.

So you just use NLTK download averaged Perceptron tagger ENG which is for English language. So you will also have data set for different languages, right?
So now if you see whatever words you saw earlier, right? The is what DT and quick is adjective. Brown is noun, then fox is noun. Jumps is a verb, but since it is using S, it is WBZ.
And then you have over there is again DT lazy and dog.
POS, as I said, POS NER stemming lemmatization is a trained algorithms and it is trained on a word data called Wordnet.
Word net and it's also using this algorithm which is averaged perceptron tagger and what language are we using for English. So this might not work well if you had Gujarati here in sentence right? So for given data set you have to use the given sentence.
So let's suppose you run NLTK 1st and when you use POS tag and if you have not given the download function, you will get.
In your environment you you are. I'm in the environment. I can't see it anywhere. So can you check your route wherever your library is saved right? You will see a path where it is downloading.
Oh, and then they get. Oh, OK, my identity get data. Yeah, it will save the files.
But all these files are not heavy weight. It will be hardly less.
Till here everyone are done. So the logic is just import and pass or input statements. If that particular algorithm is dependent on any data set, we have to download the data set.
And here we'll take one example of sentiment analysis. So for sentiment analysis you will have to train your own model, right? And once you train the model then you can use predict whether this is positive or negative. Here we have one ready made example called radar lexicon.
So if I come back to the slides and if I show the frameworks.
Here I mentioned text BLOB, right? Text BLOB is also dependent on Vader, so Vader is mainly used for sentiment analysis.
Radar sentiment.
So if you see here Vader, it's mainly used for a lexical rule based sentiment analysis tool. So they already built a model, we are just reusing it.
So you have badder lexicon.
We are downloading it and then you're using from NLTK dot sentiment import sentiment intensity analyzer.

And then the logic is same. As soon as you import anything, the first thing is you instantiate it. So how do you instantiate whatever you have imported? Just create the object of it.
This is just like how you do, right? You define a class and you define an object.
This is the same logic and here I have a text called. I really enjoyed the movie. It was fantastic right? We can try the same example. Awesome this is **** right? So then we can check what is the negative score and what is the positive score.
So I want some this product will be neutral.
I am awesome.
I am awesome then.
So if you see the scores have gone down, so we can have thresholding here.
So if you see both positive and negatively are very close by.
If if you just say this product place like if you if you with the old database like this say like this product slace this product slace.
Yes, yes, yes. OK, correct. No, it's neutral. Neutral.
This is a very positive feedback, huh? It's added in neutral.
This product is ****. It is like, you know it's still neutral. No negative is 0.59, but this is very slow like you should have threshold at least.
60 or 75%.
Slaves and hills is also positive in this time. So it is negative 0.4 power. This product has killed our customer.
Negative. OK, negative with maybe, but uh, the neutral is not that great.
So usually what we do in sentiment analysis is you have to train models, right? So one of the model that is better is Roberta.
So if you see as soon as I add Robota, it showed sentiment analysis. So this model was actually built for improved performance. So where can you use it directly from mugging face?
We don't have it, no. So basically what happens is Roberta is built by Facebook, right? OK, so you have the base model. Now you have Twitter data, so you just feed the Twitter data to Roberta model. But how? OK, how fine tune it.
But you this is fine tuning. OK, this is fine-tuned on Twitter data. No, it's uh, it's fine-tuned on Twitter data. OK, the robot model is fine-tuned on Twitter data.
So the model was I guess Robota was built by Facebook or it should have the Robota is by Facebook. So you have this model base model and once you have this base model either you can pass the Flipkart data, Twitter data, train that model.

And then use it for inference.
And if anyone is building on question and answers then you have birth.
How do you know like for what purpose which model to use GPT? So basically in our case what we usually do is there are three. If you are not going with LLMS then there are only three approaches. Is it decoder only model or is it encoder only model or is it decoder plus encoder?
Encoder right? So if you are using anything on sentiment analysis, you might not need any encoder, huh? Decoder decoder only. So either our best option is GPT 2 or it is Robota. Now how do I know which is decoder only?
So if you just search for GPT 2.
No, I can't refer GPT because I don't know GPT. This is where I said GPT, right? So the only thing what you have to keep in mind is do you need decoder only? OK, decoder, encoder or encoder only. So now let's suppose my task is take summarization then I need.
Decoder and encoder. So this is where you use T5. So now how do I know which model that we can search? OK, so the only core understanding is because we need the encoder and decoder and correct to ask like which models are best in encoder, decoder.
OK, so now if I ask GPD two, if you see his GPD two decoder only, yes, GPD two. So GPD two will be if you are using rule based ML algorithms, right? Most of the time not many people use LLMS.
OK, so during that time, obviously you'll have to go with the best models available. In that case, these models are best. OK, one is GPT 2 and if you are going with tech summarization, you have P5 for.
So if you see, you saw summarization, yeah, right.
No, this is not the base model.
Base model is from Google.
If I.
Uh, no. So this is transformers model basically.
OK, so BERT is by Google. OK, T5 Google, Roberta is uh, Facebook. So how do we use Roberta now?
Roberta will have to use from hugging face. You will have to import hugging face tokenization and this.
And this was all NLTK as of now. Uh, here we are just using the existing Uh algorithms. OK.

Is this clear? OK, so we have one more example for how to get synonyms and antonyms. So if you look at all these things, you're using Wordnet data. Wordnet is not that much huge data set.
If you compare that with GPT models or BERT models, that trained on at least millions of text tokens, whereas Wordnet it will not be much. I'll just check how much Wordnet data set.
Size.
It's not even a million, it's hardly 202 hundred K, which doesn't make sense when it comes to LLM world LLM. It's playing in billions and when it comes to BERT, it will be somewhere around millions.
And these models are in thousands.
So if I search for BERT model data set size.
It's still in 3.3 billions, which is fine.
If we search for GPD 3.5.
175 billion.
This is again very old. During this time we didn't add GPT 4, so probably GPT 4 is more than trillion itself.
So based on the data that you have, you build models in that way. So whatever you're using in NLTK, it's using this word net.
What's corpus? Corpus is nothing but data.
Collection of data.
So when we meant corpus, that means we are utilizing this data. So from NLTK dot corpus import Wordnet. So Wordnet is the data set. Now what we are trying to do is once I define Wordnet, I want to check the synonym of a word called good, right? And then you just have to do results equals to wordnet dot since sets is the function name and then you add your word.
So as soon as you run this word, you will have different.
Jason files. So if you see you have since set here basically what you need to do is you need to extract a function called lemmas. So inside lemmas you just have to get the name. So name is nothing but your synonyms and in the same.
Sentence which is synth. You have a function called antonyms.
So basically what you're trying to do is you instantiate. The instantiation is nothing but Wordnet and once you have Wordnet there is a function called synsets. In synsets you just have to pass the word that you need and this word is not giving you a direct results, it's a list.

And once you run it, now what is this insert? What do you mean by this thing?
If you see something in capital, so this is an class. So what I'm trying to do is huh from inside this object. Now what is this sin? Sin is this line.
So from that what you need to do is there is a function called lemmas. From there if you are getting the name that is considered as synonyms. Is this clear? You are getting an object. Inside that object you have a function called name.
That name belongs to synonyms, and if you do lemmas dot antonyms, let's suppose if that word as the antonyms, then you just have to extract antonyms zeroth index.
So if you're still confused on how this happens.
Just comment this code.
Print it out. Just print. What is it generating?
And then if you see it again, what is it? What is happening here? It's again a class, right? So again, what you're trying to do is you have to loop because there is a list.
So now I'll just comment this print and then I will just use a function called lemma name.
So lemma dot name.
Hi even you can do that if you do DIR lemma.
But here what you need to do is you just start with break here. Why? Because it will print same thing multiple times.
Why 2 breaks? I want to stop both follow-ups.
So you have antonyms.
And you have multiple functions as well.
What we need is new.
Again, documentation is best choice just to see which function to use and I will just use dot name. So name is a if you see this.
Of Q that means it shows that. So you can't use an attribute. The attribute will be a different sequence.
So since name is a function, I'll have to make it as a function.
OK, why am I printing this?
So now we say it shows good if I remove break from here and also break from here. You'll get all the other words, but what I need to do is I need to remove the good one, right? So here what we'll do is.
Synonyms append and then I'm checking if it has antonyms. If it has antonyms, I'll just get the antonyms 0th index and name and then I'm just appending it into a list.
Is this clear? Obviously you won't use this code anywhere, but you will have different

logic that you can utilize. The only thing is you have to use bigger data set then.
What word did you give? Synsets.
S.
So now I can give any other word as well boring.
So the synonyms of bored is bore, boring, then drill, drilling, then slow, tire, tiresome and antonyms. It's interest. So most of the time antonyms you are getting direct, but for synonyms you are getting bunch of other details.
If I give interest.
I'll remove this.
And for interest, it shows bore and uninterestingness.
Is this clear? These are just the functions that we are utilizing.
And now when we look at NER for NER, if you see you have average perceptron tracker which is already installed.
It is like we did it for POS, right? POS tag. Where will we actually use it? POS tag directly will not use it anywhere. So basically POS tagging is used in the tokenization part which is internally happening in most of the frameworks now. So this is just core knowledge.
Of NLP which is not. You will not use it directly now in most of the frameworks because it's already inbuilt. Mainly it will come in tokenizer part because tokenization happened 1st and then.
You will not see any tagging. No, no. I'm saying like Intel would work on post tags. It would remove the and the yeah, yeah, yeah tags will be removed.
So one thing is for embedding it works because for embeddings let's suppose I tell king and queen. So how do I know what king is and what queen is? So their tags are used so that we can get this course schematics course right? So you have king.
You have queen and then you have soldier. You have ors. OK, let's take 8 words king.
Uh, let me open Excalibra.
So this is a vector space, right?
So here you have.
Let's suppose in some vector space you have king.
So somewhere close by you'll also have Queen.
OK, now you have soldier. Somewhere close. You'll also have soldier. Now you have ors.
So if you look at this keywords, king, queen, soldier or matches right, all the things are referred in one of the context. Now if I bring cat and dog, so this will be in

animals, right? Cat is an animal.
Then somewhere close by you'll have dog.
And in some vector space you'll have, let's suppose lion.
Or lion. Probably I can have somewhere in between as well, which is 1 animal commonly used in no, I mean in royal kingdoms.
Huh. So somewhere I'll add elephant.
So now if you see, let's suppose you're building a search algorithm which is used for vector search, vector search and schematic.
This orse can be categorized inside a vector space of animals. If not, it can also be categorized when something is happening in the royalty, right? So every single words will be grouped in a vector space and they have schematic scoring.
So based on this course you rank the user query. So if I ask any question related to my resume, it will do search and get a scoring results right here you're only playing with vectors.
Right for vectors you have embeddings in embeddings.
You're using schematic, schematic analysis.
Once vectors are done, then you will find the distance between them.
So in distance is where you calculate the similarity.
Between.
Two vectors. So where is this used? Let's suppose recommendation system. I like one piece.
Right. I'll I'm going in some direction and then there is one more person who likes Naruto. He will be in One Direction. Then there is someone who will dislike this, right? So when you calculate this distance measurement, right, you have cosine similarity.
So what is cosine? If both of us have similar interest, we are in positive degree which is 90. And if both of us are in different direction which is 180 degree, that is opposite -1. There is no similarity between me and you. So that time my recommendation system will not recommend you or this thing.
So first you calculate vectors. Vectors is simple like it will be like.
0.40.30.9 some values then distance measurement. You will apply some algorithm and you will find between those two, which we have in probably two more days. This vector calculation you're training a model.
So their schematic analysis in old school you had POS tag. You will not use it directly, you're directly using the.

Already developed embeddings, for example open AI, Gemini.
So usually we'll not build this models, you're just reusing it.
So again, whenever we are starting with any function in NLTK, there are some of the existing data set that we need to download.
Here I have some sentence I have. I'm the hope of the universe. Then you have some quotes, right? So once you define these quotes, how do I define POS tag? What is the logic? If I want to get the POS tag, what was the first step?
We have a sentence after sentence tokenize after tokenize POS tag. So if you see first I want to get the tokens. Once I get the tokens I'm passing it into POS tag and then print the POS tag. We already did this right?
Now if I run it I for I what is that uh POS tag? This is same what we have done. Now what I'm trying to do is I want to get NER. What is NER named entity recognition. So in order to find NER.
You have to again download some package which is chunker, then words and maxnet any chunker.
So how do I know what I need to download? Just run that particular function. Let's suppose I run any chunk. If I get lookup error, this name will be there. All these names.
Right, so 90% of the time we'll get these errors look up error, right? Whenever you use NLTK, let's suppose I want to build synonyms and antonyms, I will get an error saying hey you don't have this package. So I'll just copy that look up error download method, come to a different file, run it and then use it in my main dot PY.
So always whenever you're working with NLTK.
Nowadays we have better library, so very rare senses. If we come across NLTK, first thing is have a file called download dot PY and then I have main dot PY.
Never use.
NLTK dot download.
In your main inference file.
What happened? So it's always better to have it in a different file.
Because every time let's suppose I deploy main dot PY in streamlet right, it will do NLTK dot download every time the user refreshes the page, which is not the ideal way to do right? So we run it once. Once it's deployed then we can use it in main dot PY which is.
Hey in my development environment I already have this data saved. You just have to use main dot PY. Is this clear?

And here also, right, let's suppose you use three different functions. Only I have synonyms, I have antonyms, and then I have lemmatization. 90% of the time you will only use lemmatization in NLTK.
Other things like sentiment analysis, no one will use it. You will use an ugging face. Anonyms and synonyms also will use ugging face. Only lemmatization is supported in NLTK. There is no other libraries where you can use lemmatization. So only for lemmatization there might be 3 downloads.
And if you're using POS tag as well, you'll have two downloads. So during that time just comment some of the already downloaded so you don't have to download it again. Even if you download it's fine, you will get true or false right? So but the only thing is don't have NLTK downloading main dot POI.
That's it. And here I have some sentence. Barack Obama was born in Hawaii, so this should be PER. This is O, this is O, and then Hawaii is.
Awai should be what GPE? And then if you see Apple, this is confusing when it comes to Apple, right? Either it can be a organization or it can be a fruit. If it is fruit it is. Nothing. I mean, noun is there, but in terms of named entity recognition, Apple is O. So what does it do? It's taking GPE.
OK, GP which is not at all makes sense. It should be organization right? So this is what happens with what you call lesser data set train model. So if you see Barack Obama person, person O will not be printed GP for a while.
What else do we have in second sentence UK?
UK does not printed anything. Elon Musk person person SpaceX. It has added organization California GP. So for some of the important keywords it has just missed it. If I make this as data, nobody would know X.
Founded X. It got a very correct. It's X in the identification. It does, but X no, you're saying X.
So if I run this only XI don't think it will recognize. It does not have that data line. It is not trained on that. X will be. X is ignored. Border line does not identify. What? What is the? What is the LOC? LOC might be.
Hello, Steve.
Which one?
VPA VPA. They don't know the new. They will also VPA OK.
OK, repeat. OK, what about like the the post code later, which is like close and not running the old data set. Yeah, the like.
Vimal can come in PR.

Satyam OK, Satyam it has given GP.
So if I you Vimal, Vimal or who is person, who is person.
Vimal it as a reddus person.
So these are the I mean so far whatever you saw probably you have different libraries to do it. But when it comes to preprocessing, usually people just use an LTK and for preprocessing you have stopwords, right? Lemmatization stopwords.
Is where you'll actually use NLTK, right? So NLTK dot download stopwords and where are stopwords used in recent days?
Uh.
How many of you know hybrid search in RAG? Have you ever attended any events?
No. OK, I'll skip this for time being. I'll show this when we cover vector databases, but I'll just show quadrant tech documentation.
Concept.
OK.
OK.
So basically in vector this thing also right everyone of vector databases. So what does vector database do? So you upload your data somewhere and then you ask any question. It will look into database and it will retrieve anything. So in payload you can also define if anyone gives this keywords which is stop words, don't even search it. Right, so you can avoid some kind of memory optimization. So this was never there in vector database until now, but in recent releases, vector databases like Milwus, PV8 and Quadrant, they've added stopword support right, which is mainly added in payload, but this is something that we will cover.
When we discuss vector databases and second thing is you have hybrid search, so let me check if that line is visible.
OK, it's been 930.
OK, can you see this files? So if you see when I'm whenever I'm defining vector databases with hybrid search approach, hybrid searches combine vector search with keyword search, it will download some files.
Arabic dot text, that's dot text, Spanish dot text every single languages it has stop words right? So this stop words needs to be removed and also it has text supported data in this. So whenever you are defining stop words, it's not just in English language, it will be in different languages as well.
So we just have to define which language we need. By default this stop words whatever we have right? If you see you have stop words dot words English.

So here you can also choose different languages based on what data you have.
So let's quickly wrap this. But if you look at the logic, it's the same. Whatever function we are using, does it have any look up error? If it it has, copy that, run it in a different file and then just do instantiation.
So now if I print stop words.
It has the, their, them, themself. All these are considered as common words, right? Common words needs to be removed from the.
Data before you pass it to the model.
So here I again I have the same codes. What I'm trying to do is I want to loop every single sentence and I want to remove stop words from them. So I will remove. I mean I'll loop through the entire sentence, convert into tokenize if that particular word is not into the stop words.
Then I will append it. If it is there, I won't even add it. So what is this?
What is this?
What? What logic is this? This is list?
Pomper.
Is this clear the logic? What is happening? You're running a loop and then you're checking the condition. This is the entire if condition. If it is not in the stop words, then only append it.
If it is in stop words, it will ignore and then you are just checking what was the original token and the filtered one.
So if you see I am the was remote of the was remote and here I am the was remote to be the was remote and also of the was remote.
Is this clear? Stop words?
And again, when it comes to stopwords, right, you have to define based on your business statements or use case. If you think it's just sentiment analysis, then you can remove stopwords. But if you're working with next word predictor, you can't remove stopwords.
Because in next word predictor, stop words are very important because people will just add stop words, right? They won't search high technical words. So based on the use cases, you need to define whether you need to remove stop words or not. For sentiment analysis you can remove.
For language translation you can remove because for language translation do you think stop words matter? No, you can remove. For text auto completion it matters, so you won't remove. So based on the use case you have to decide what kind of pre-

processing to do.
And this is the last topic. We have Porter stemmer which is for stemming and I have some of the statements. What will playing be?
Play. Played. Play. Happily.
In semi you can't define, but technically it should be happy flies. It should be fly, then better. It should be good, but it will be better best.
Best will be best running, run, runs, run.
So the logic is again same. First import it, then instantiate. Once you instantiate, use that function. What is the function stem?
Is this clear?
So you have play play. Then this is grammatically wrong, but stemming will do that. FLI better best run done. Now lemmatization logic is same. If I run these two things I should get an error.
Because here I technically wanted to show look up error.
Okay, probably it might already been added.
Let me so this one is a right this line. This line will be asked when you're using lemmatization.
So what is the logic from NLTK dot stem import word lemmatizer, instantiate and use that function. What is the function now? Lemmatize. No lemmatize.
So these are the only two functions you need to remember. One is stop words and one more is lemmatization. The other things are just to show you the format, what NLTK does and what is the format of it. Download instantiate user function. That's it.
But does it also handle I would say also handles spelling errors? No, no.
Yeah, standardization mostly the same. No, no. So if you see fly, fly, appily is appily, fly, fly, then fly. Why appily? Obviously there are other words as well.
Here, appily is what?
It is more like it makes a sound lies flying. It just cuts off the ES does not take care of.
So here what is happily probably it is adjective right? So here if I give.
Adjective it is hard.
Or is it DA?
You're in for now, as per at next.
OK, didn't remove. Probably it's V only.
There is R for now, no N is for now.
N is for noun. Yes, V is for verbs, A is for adjectives, R is for adverbs.
It didn't remove, so it should be V.

The idea is we the standard.
Is this clear? Usually you'll only use lemmatization and stop words. The other things you have better things in library. I mean hugging face like synonyms, anonyms. You can use some better algorithm sentiment analysis. We can use Robota.
And what else did we had here? POS tagging. It's already in most of the embeddings, so we'll not use it. So for embeddings, if you are going with closed source, it is open AI. If not, we use NTB leaderboard.
If you're using an open source, but we'll cover this in coming days. We took too much of time today.
So here if you see, you'll only find open source LLMS, sorry, open source embedding model.
You have when 3 when three Gemini embedding. I guess I made it open source. Then you have you should see somewhere GTE Gina embeddings is also very useful. We'll be using Gina embeddings in some of the.
Examples when we use fast embed.
But mostly you guys will use Open AI because it's cost effective and also better than all these models.
But Gina 3 is there in one of the example.
Oh, any questions from today?
So Numpy pandas we will cover in online for in person we'll only cover NLPNLP related concept.
Oh, should I exit in this Microsoft Meet? Yeah.

Tirth stopped transcription