

# Developers

## NLP as an API

Text classification using Keras,  
FastAPI and NoSQL

Based on original content by [CodingEnterpreneurs](#)



# 01

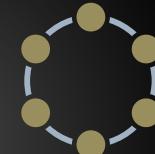


## Housekeeping Live and Hands-on



# 02

## Our Goal Use case and Technologies



# 03

## Database Setup Data model & Astra DB

# 04



## AI Train a text classifier

# 05



## API Bring to production



# 06

## What's next? Quiz, Homework, Agenda





# Housekeeping

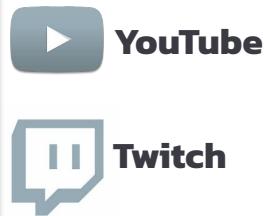
Live and Hands-on



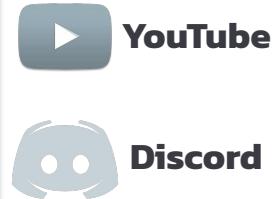
**Livestream:** youtube.com/DataStaxDevs

**Questions:** <https://dtsx.io/discord>

#### Agenda



A screenshot of a Discord channel titled "workshop-chat". The channel has over 100 members. A user named "isole001" asks if it's ok to use the workshop background image. Another user, "Erick.Fernandes", responds that it's fine as long as it's not copyrighted. The channel also contains messages about DataStax Astra and reactive drivers.



## Games and quizzes: [menti.com](https://menti.com)

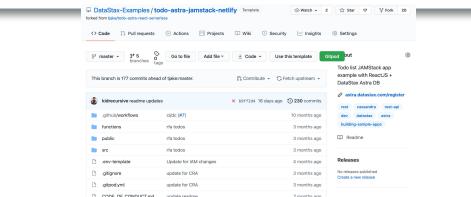
How much experience do you have with the Spring Framework ?



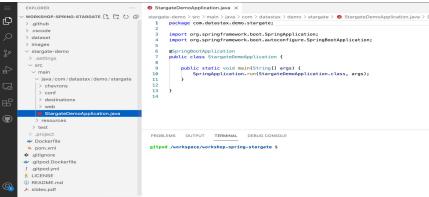
Live Sessions

Nothing to install !

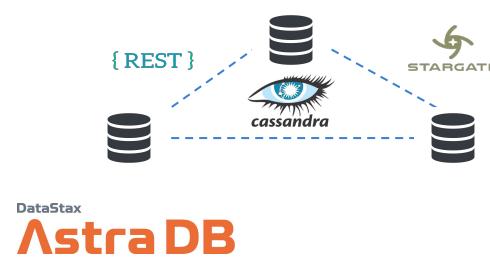
### Source code + exercises + slides



### IDE



### Database + GraphQL + PlayGround



DataStax  
**Astra DB**



**CodingEnterpreneurs**

Youtube & blog for more

Hands-On Housekeeping

# 01



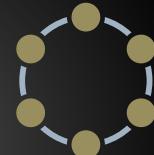
## Housekeeping Live and Hands-on

# 02



## Our Goal Use case and Technologies

# 03



## Database Setup Data model & Astra DB

# 04



## AI Train a text classifier

# 05



## API Bring to production

# 06



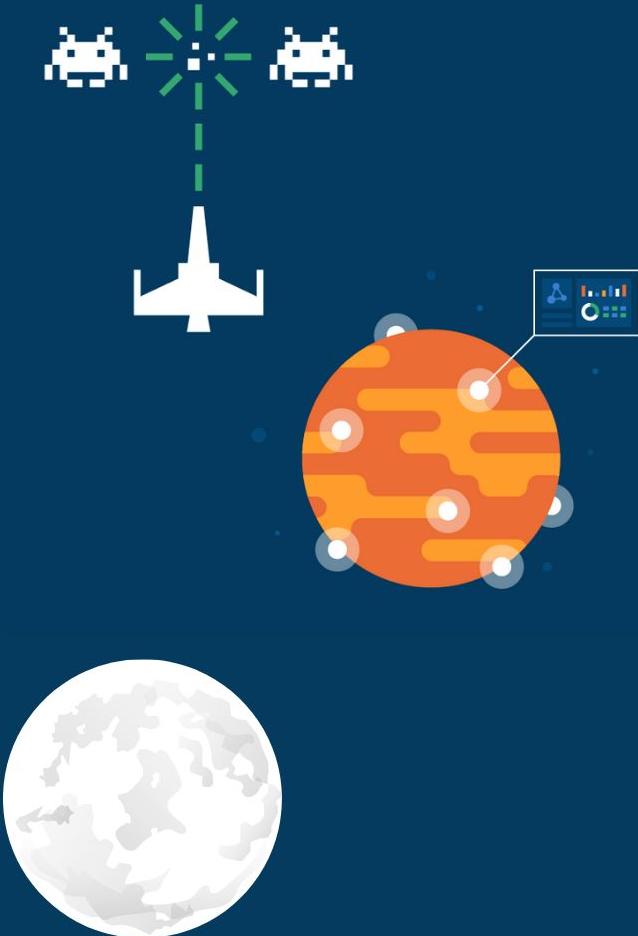
## What's next? Quiz, Homework, Agenda

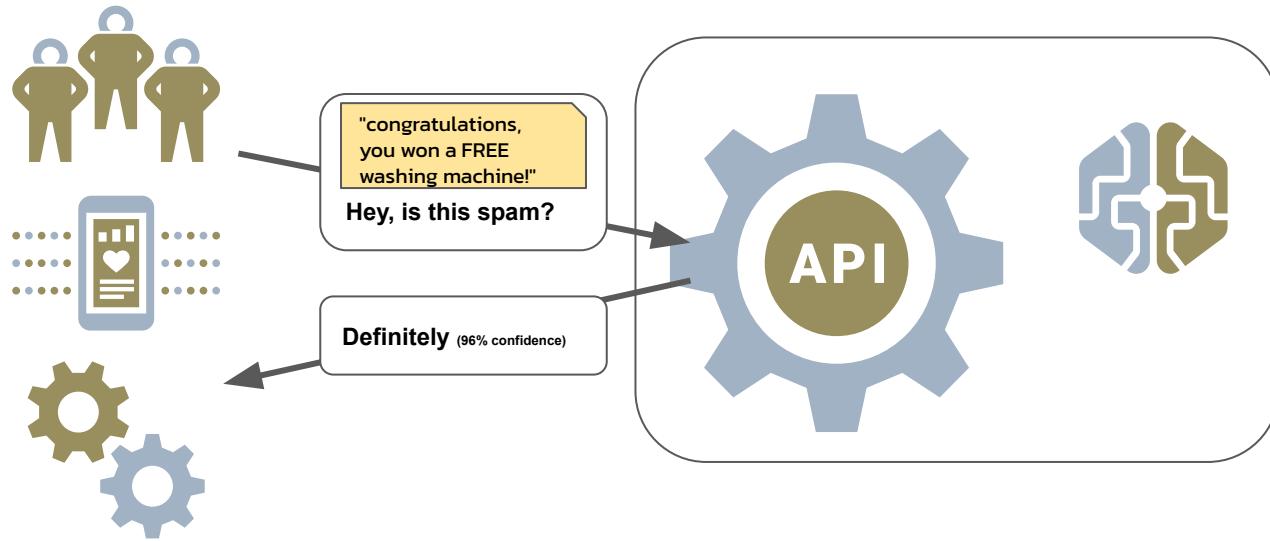




Our Goal

Use cases and technologies



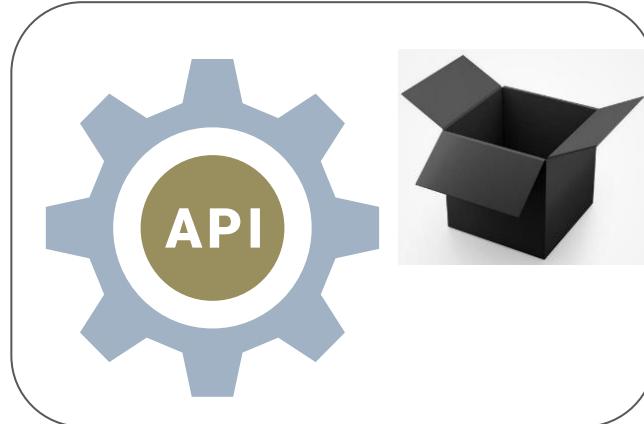


What we want

The API sees the AI part as a black box

API = concerned with stability & performance

Engineer Stuff™



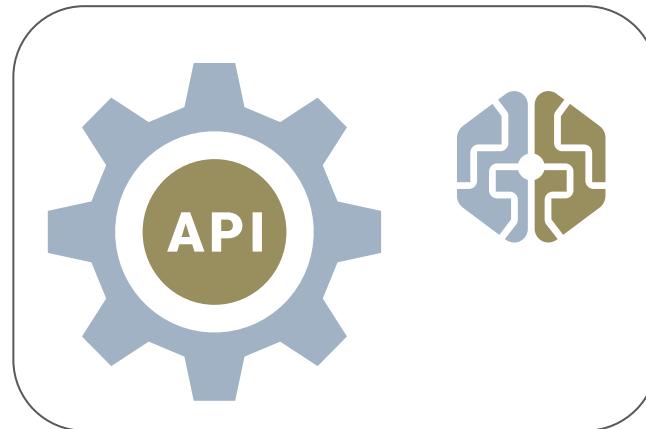
Traditional viewpoint



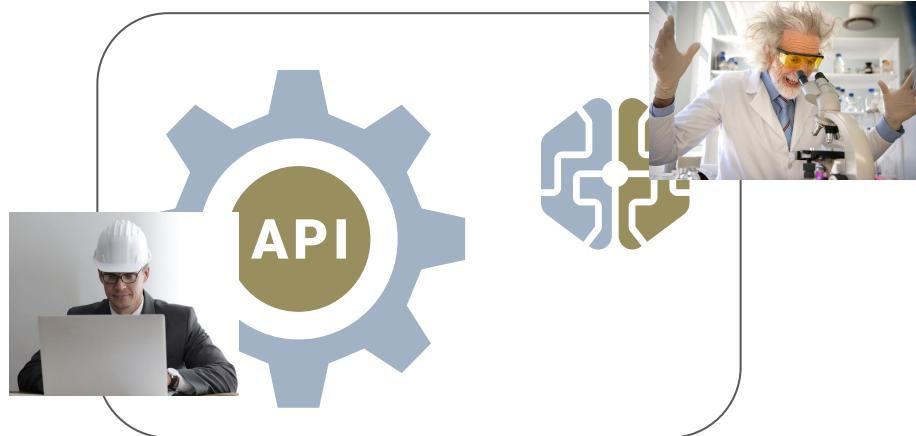
ML components, often messy  
(no offence meant)

(hardly reproducible) treasures  
hidden away

A "science" proper  
(experimentation, prototyping...)



Traditional viewpoint

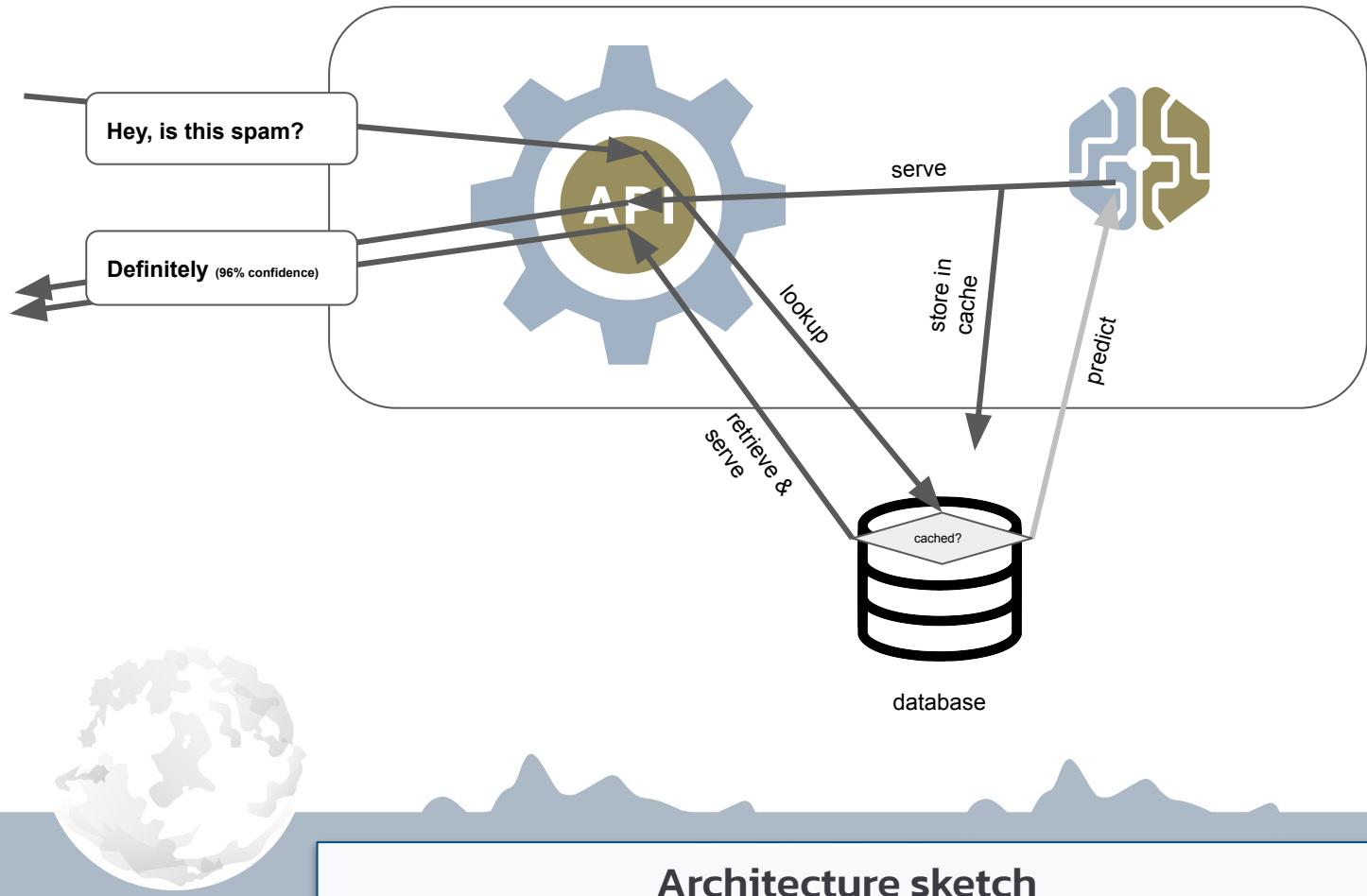


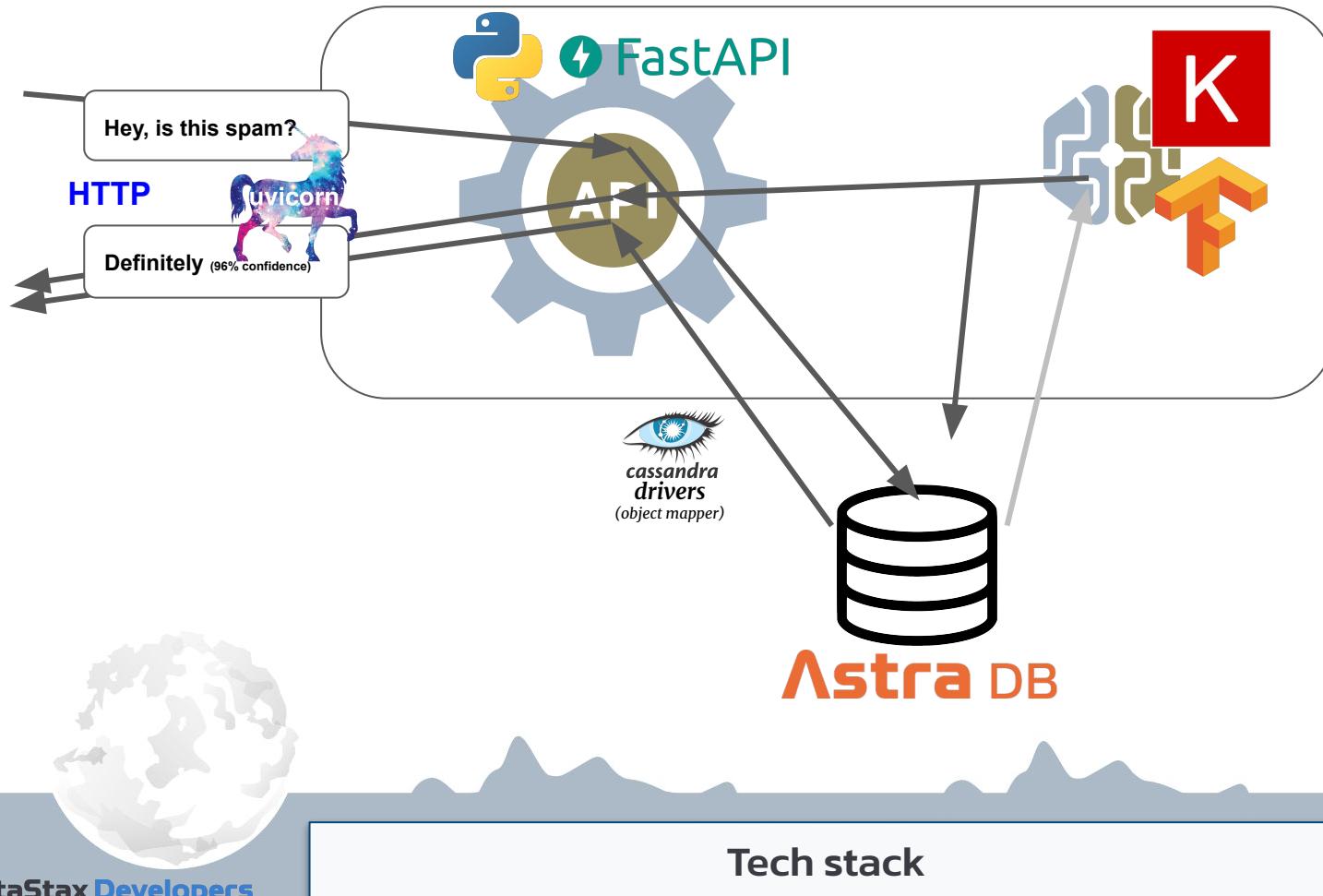
Properly exposing ML elements  
in production is a key skill

Still, two clearly separate phases:

- create the classifier (design, train)
- bring it to production

Two in one





# 01



## Housekeeping Live and Hands-on

# 02



## Our Goal Use case and Technologies

# 03



## Database Setup Data model & Astra DB

# 04



## AI Train a text classifier

# 05



## API Bring to production

# 06



## What's next? Quiz, Homework, Agenda

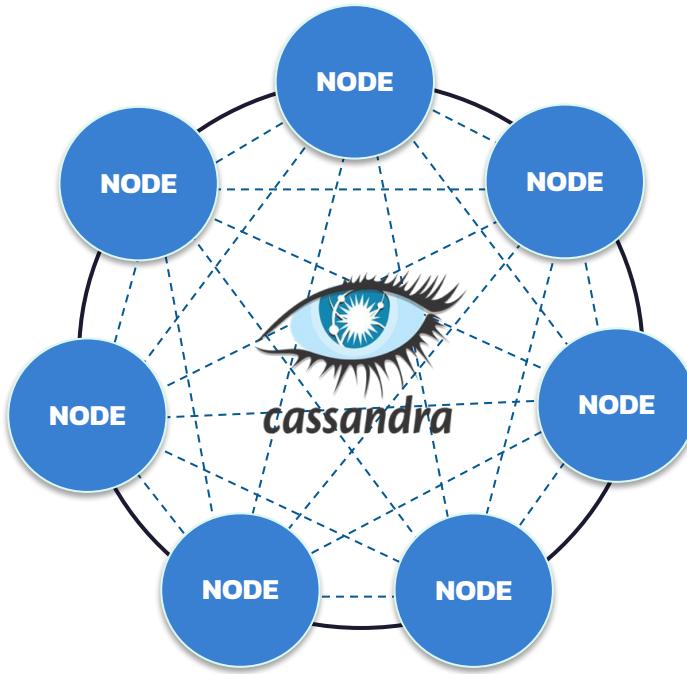


# Database setup



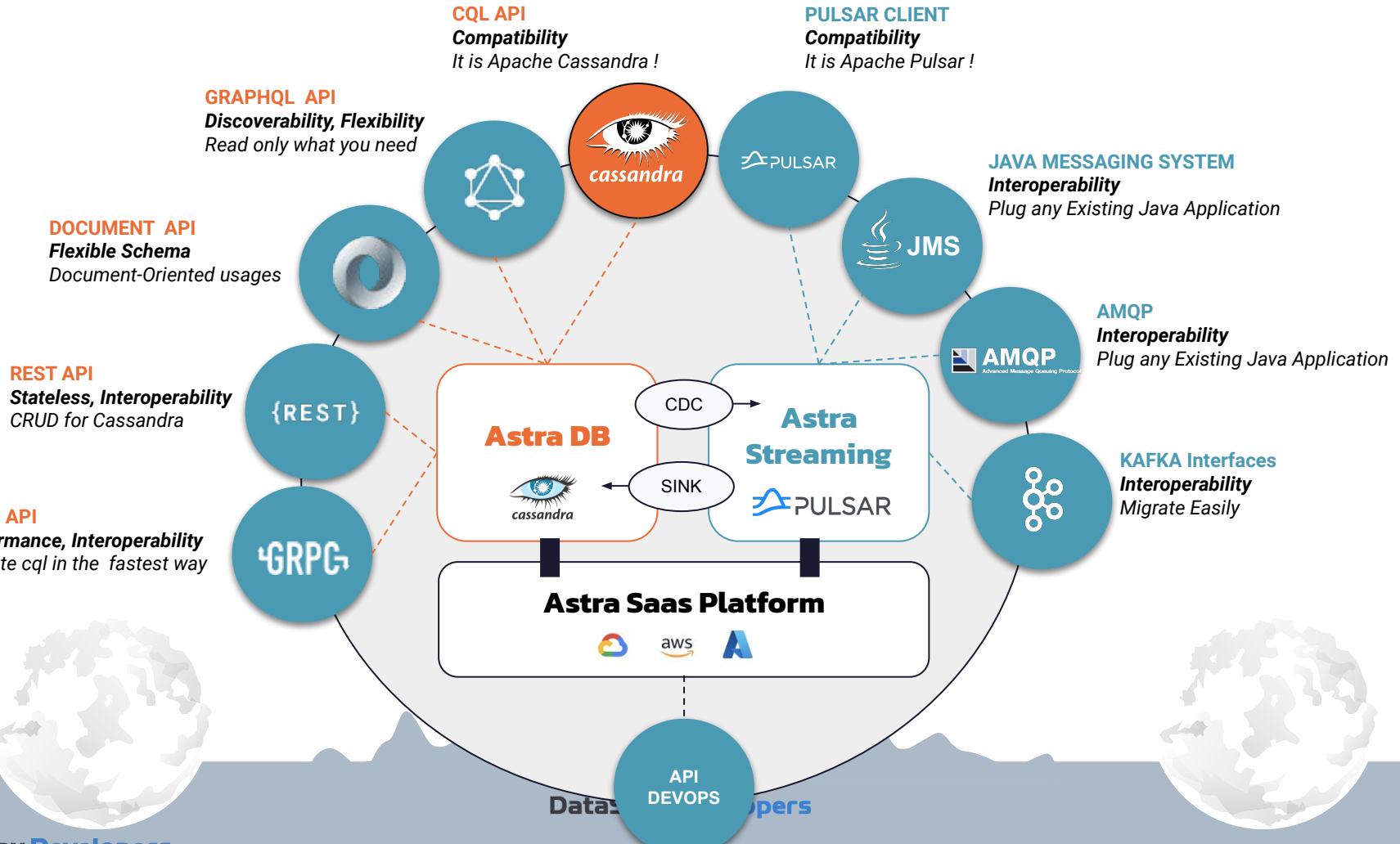
Data model & Astra DB

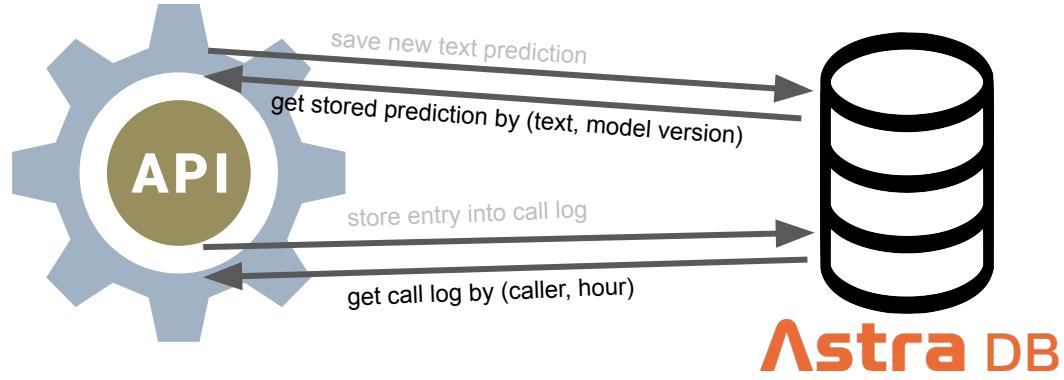




- Big Data Ready
- Read / Write Performance
- Linear Scalability
- Highest Availability
- Self-Healing and Automation
- Geographical Distribution
- Platform Agnostic
- Vendor Independent

Introduction to Apache Cassandra™

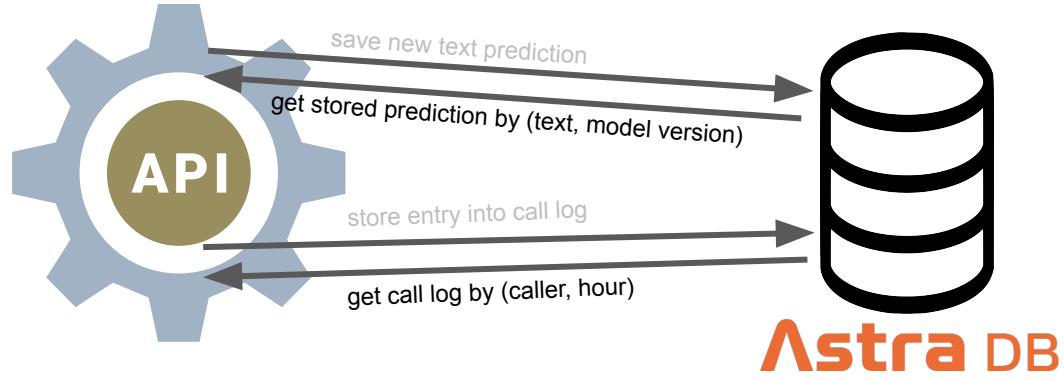




Data modeling, the Cassandra way:

***"design the table after the query"***

(also: tables are partitioned!)



Astra DB

## spam\_cache\_items

model_version	text	K
input	text	C↑
confidence	float	
prediction_map	map<text, float>	
result	text	
stored_at	timeuuid	

## spam\_calls\_per\_caller

caller_id	text	K
called_hour	timestamp	K
called_at	timeuuid	C↑
input_text	text	

"partition key" (rows stored together)

"primary key" (row uniqueness)

"Chebotko diagrams"



```
class SpamCallItem(Model):
    _table_name_ = 'spam_calls_per_caller'
    _keyspace_ = ASTRA_DB_KEYSPACE
    _connection_ = 'my-astra-session'
    caller_id = columns.Text(primary_key=True, par-
    called_hour = columns.DateTime(primary_key=True)
    called_at = columns.TimeUUID(primary_key=True,
    input = columns.Text()
```

**Tables are created automatically...**

```
query = SpamCallItem.objects().filter(
    caller_id=caller_id,
    called_hour=called_hour,
)
for item in query:
    yield item
```

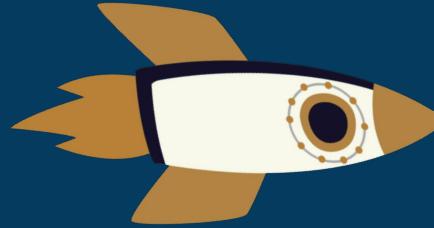
```
# Database
logging.info('          DB initialization')
DBSession = initSession()
sync_table(SpamCacheItem)
sync_table(SpamCallItem)
logging.info('          API Startup completed.')
```

```
called_hour = getThisHour()
for input in inputs:
    SpamCallItem.create(
        caller_id=caller_id,
        called_hour=called_hour,
        input=input,
    )
```



**DB I/O in practice**





# Hands-on (!github) DB SETUP

- ✓ Create your Database (or resume)
- ✓ Create a token (if needed)
- ✓ Download secure-connect-bundle

# 01



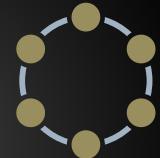
## Housekeeping Live and Hands-on

# 02



## Our Goal Use case and Technologies

# 03



## Database Setup Data model & Astra DB

# 04



## AI Train a text classifier

# 05



## API Bring to production

# 06



## What's next? Quiz, Homework, Agenda



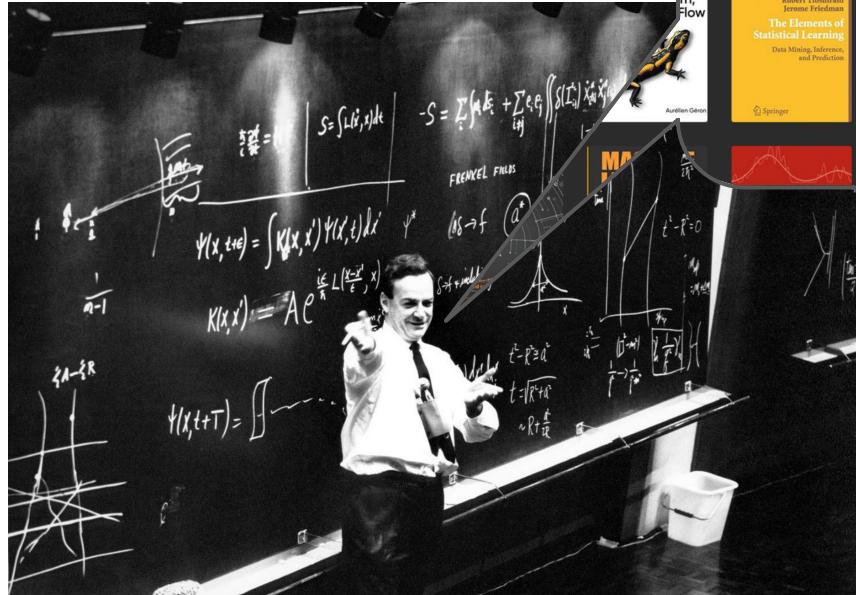


Train a text classifier



# Disclaimer

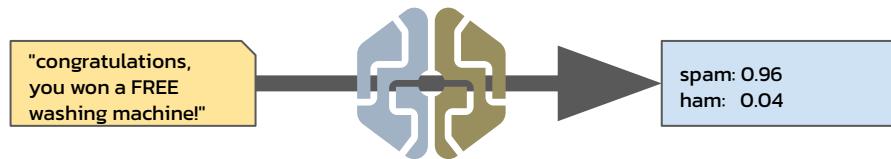
What this section is not:



Disclaimer

# "ML: LSTM RNN for NLP"

This is what we want:

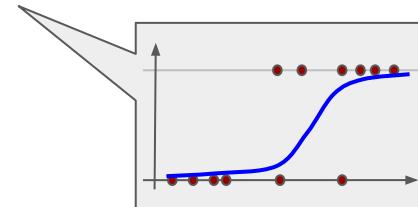
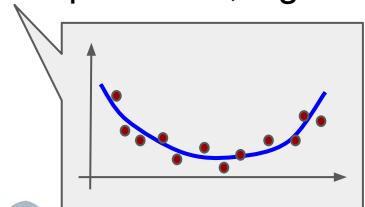
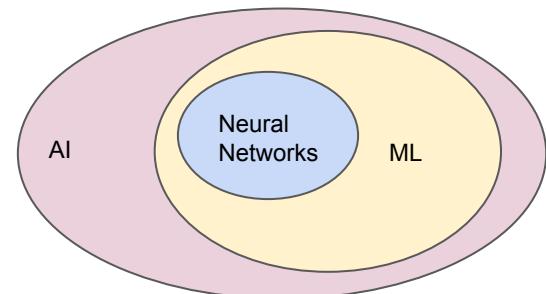


**Machine Learning** = *algorithms that improve by being fed data, without being explicitly instructed what to do.*

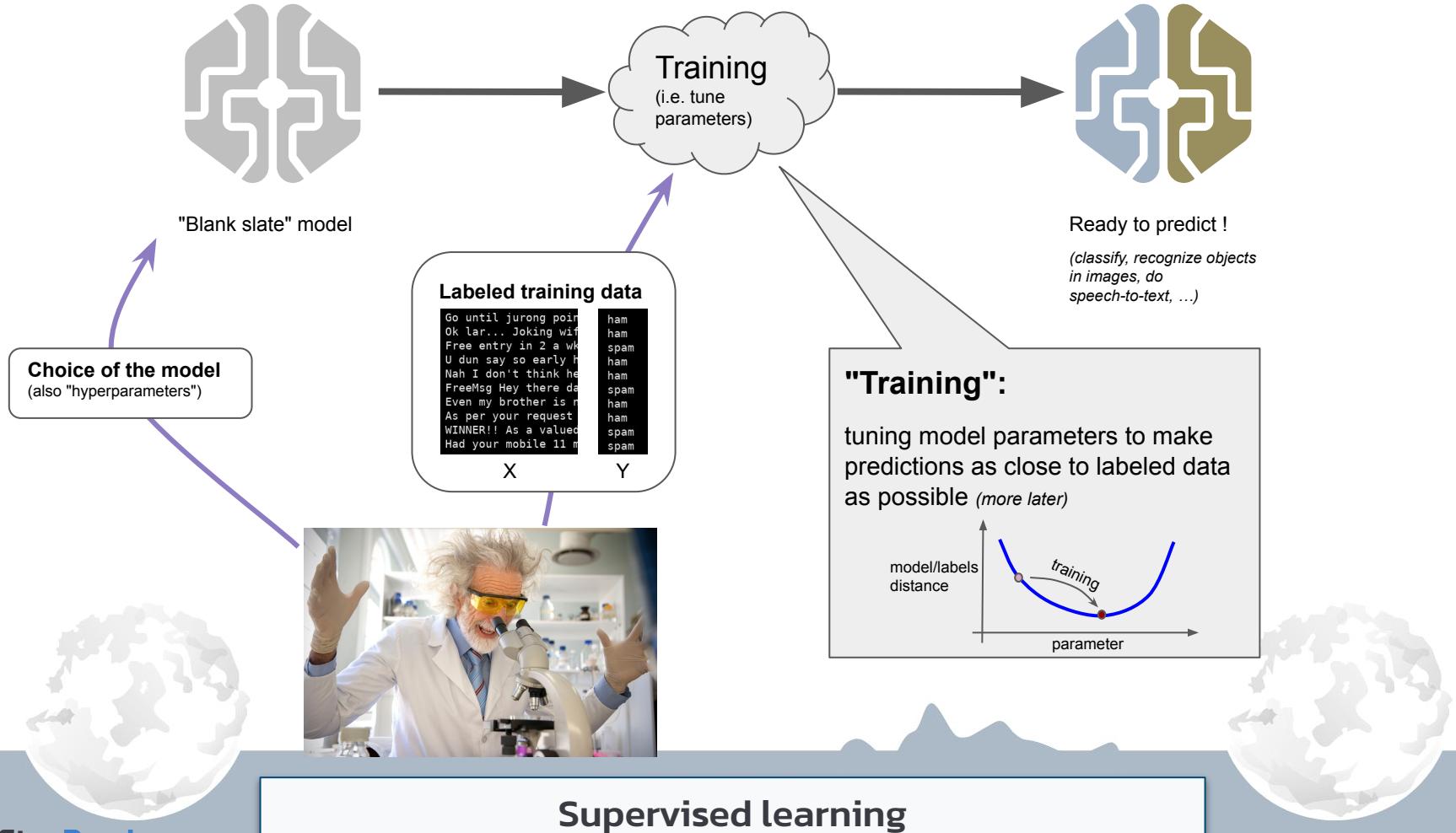
It's essentially statistical inference (with superpowers).

Lots of **math** involved (linear algebra, calculus, prob./statistics).  
Nowadays accessible as neatly-packaged tools (good for us!)

*Simple* examples of ML: least-squares fits, logistic regressions.



AI and ML



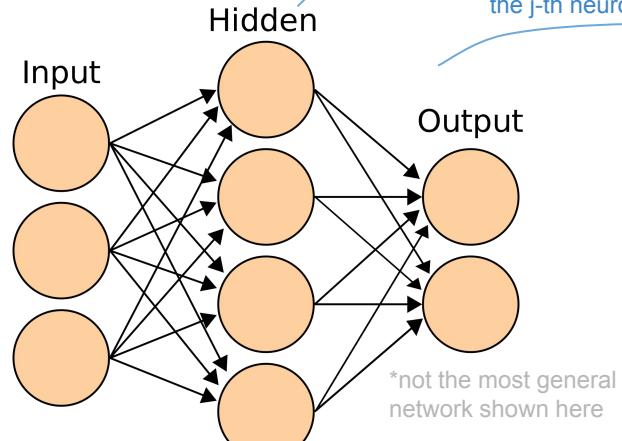
Many algorithms and structures for ML:

linear/logistic regression, support-vector-machines, decision trees/random forests, gradient boosting, ...



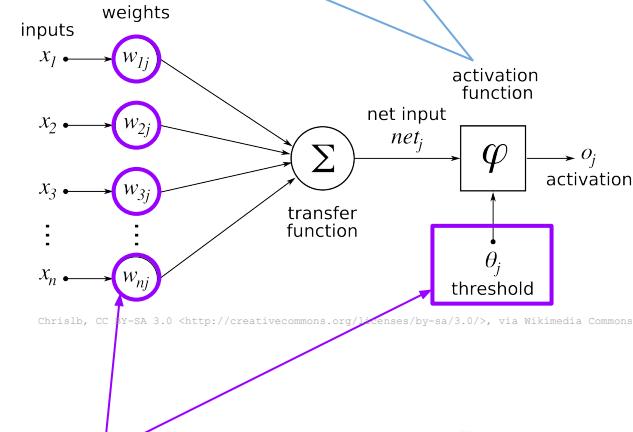
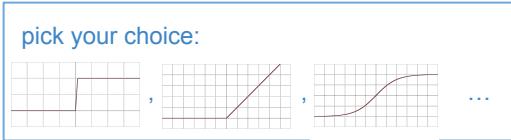
## Focus on Neural Networks

Input (text, image, ...) is encoded and fed here



By en:User:Cburnett - Own work. This W3C-unspecified vector image was created with Inkscape  
.., CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=1496812>

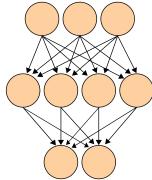
## Neural Networks



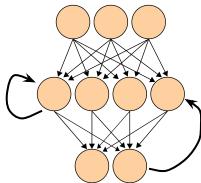
Chrislh, CC BY-SA 3.0 <<http://creativecommons.org/licenses/by-sa/3.0/>>, via Wikimedia Commons

many parameters per neuron:  
training gets CPU- and time-intensive

## Architectures



"Feed-forward"



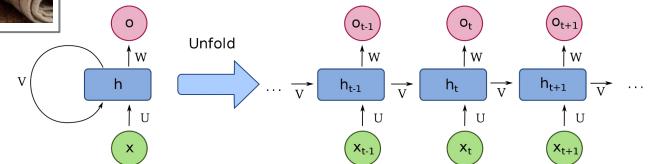
"Recurrent neural network"



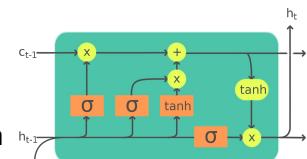
has *loops*: ~ "memory"  
better with sequences (time series, speech, **texts**, ...)  
loops "unfold" as time-steps: training gets harder

### "Long-short-term-memory"

introduces explicit memory elements  
overcomes the "vanishing gradient" training problem  
a standard RNN to process "**input sequences**"



By fdeleche - Own work, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=60109157>



By Guillaume Chevalier - File:The\_LSTM\_Cell.svg, CC BY-SA 4.0, <https://commons.wikimedia.org/w/index.php?curid=109362147>



## RNN and LSTM

# Natural language processing

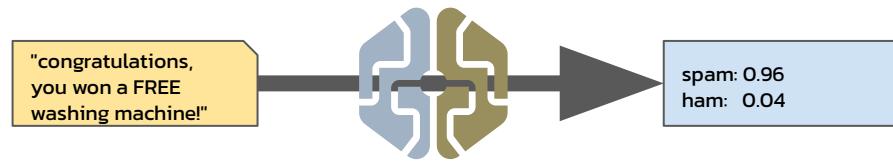
A very wide field of application for ML (since ... forever)

translation, summarization, classification, clustering, autocomplete, generation, chatbots, sentiment analysis, ...

Most of the data "out there" is unstructured (e.g. **text**)



LSTM very apt at many of these tasks (memory along sequences)



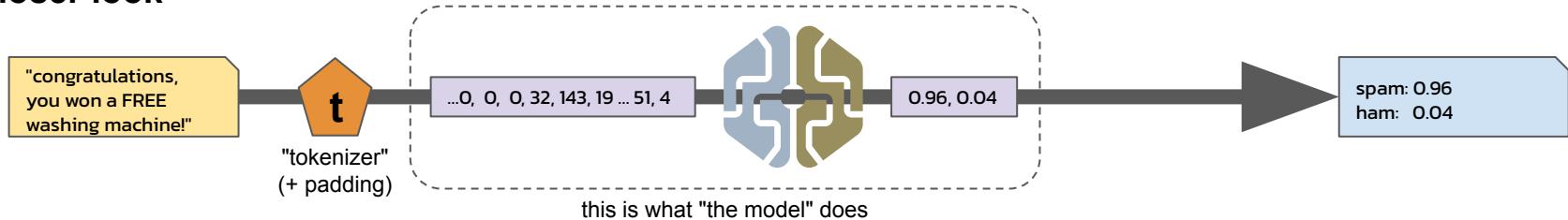
An evolving field (Word2Vec, ca. 2013; transformers e.g. BERT, ca. 2018, ...)



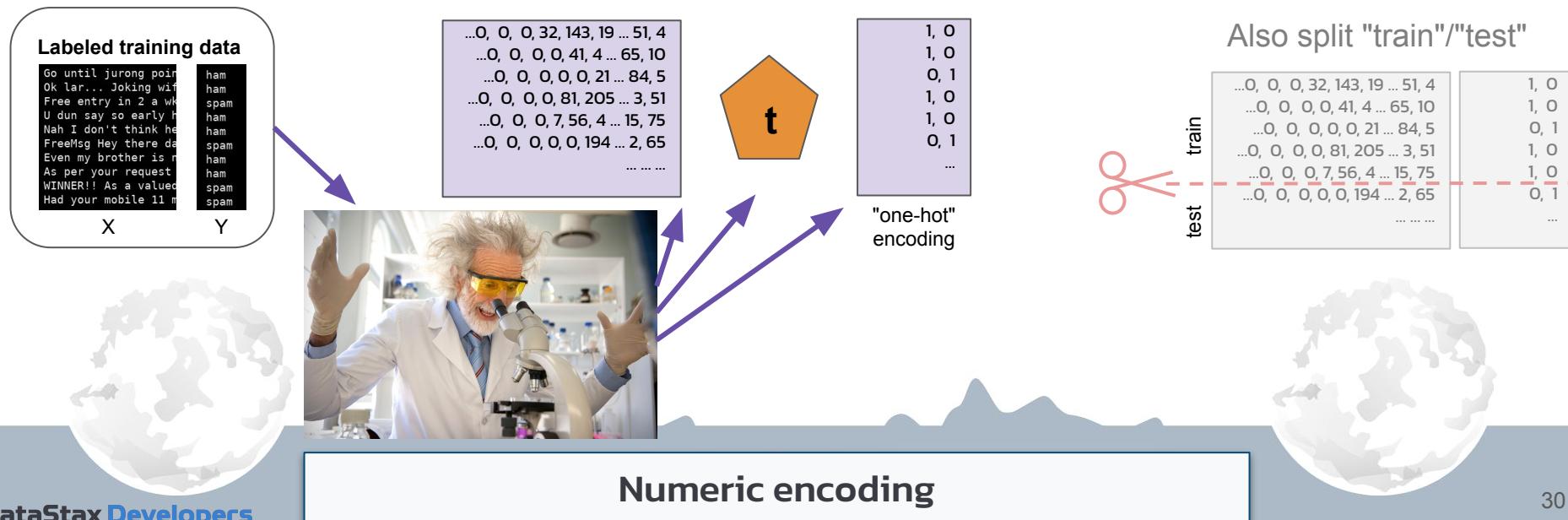
NLP

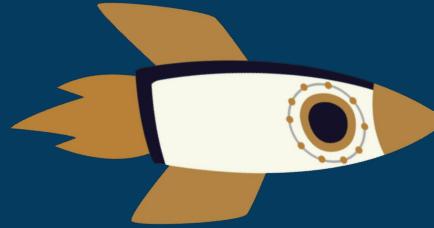


## A closer look



We must then *prepare the dataset* before training (including building the tokenizer):





# Hands-on (!github)

## TRAIN THE MODEL

- ✓ Prepare the dataset for training
- ✓ Train the NLP model
- ✓ Test the saved model



## A stack of layers:

```
model = Sequential()  
model.add(Embedding(maxNumWords, embedDim, ...))  
model.add(SpatialDropout1D(0.4))  
model.add(LSTM(LstmOut, dropout=0.3, recurrent_dropout=0.3))  
model.add(Dense(2, activation='softmax'))  
model.compile(loss='categorical_crossentropy', optimizer='adam')
```

one translates an input number into a vector

one randomly disables pieces of network to enhance training ("dropout")

then the LSTM (recurrent *within the layer*)

and a final ordinary layer reducing it all to a 2-component output (spam/ham)

**"finalize" the model: ready to train!**

inputs

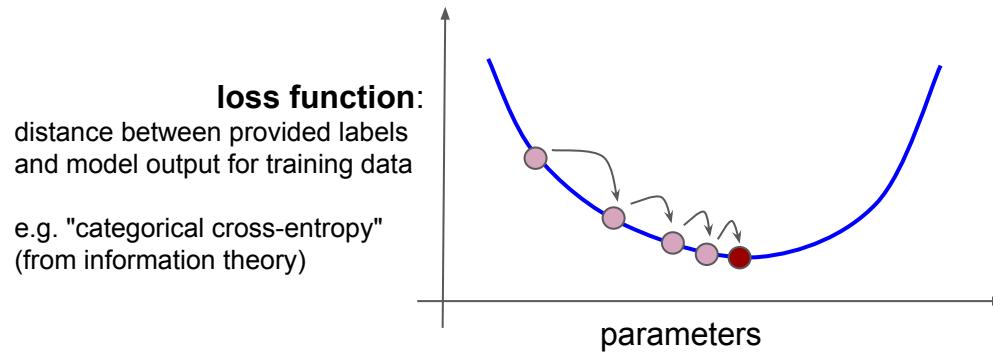


The antispam model architecture



## "Training":

(progressively) tuning model parameters to make predictions as close to labeled data as possible



### The reality:

lots of parameters, it's a mess!

```
=====
Total params: 291,034
Trainable params: 291,034
Non-trainable params: 0
```

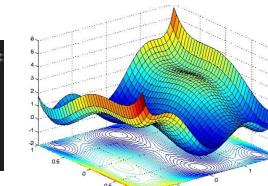


"Stochastic gradient descent" (SGD)

Backpropagation (glorified chain rule)



Drop-out



During training ...

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
```

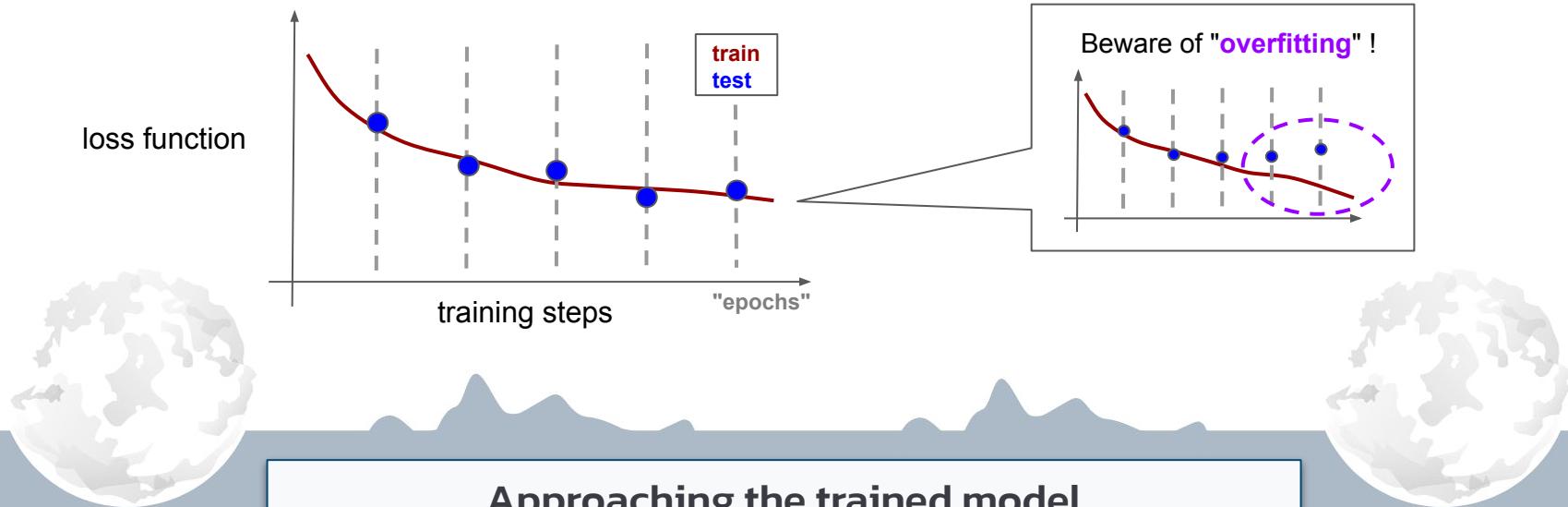
We split "train"/"test"



train	test
...0, 0, 0, 32, 143, 19 ... 51, 4	1, 0
...0, 0, 0, 0, 41, 4 ... 65, 10	1, 0
...0, 0, 0, 0, 0, 21 ... 84, 5	0, 1
...0, 0, 0, 0, 81, 205 ... 3, 51	1, 0
...0, 0, 0, 7, 56, 4 ... 15, 75	1, 0
...0, 0, 0, 0, 0, 194 ... 2, 65	0, 1
...	...

**Train on part of input dataset,  
periodically validate on *another portion* of it**

*(independently verify we're actually learning the right problem)*



# 01



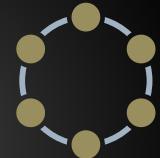
## Housekeeping Live and Hands-on

# 02



## Our Goal Use case and Technologies

# 03



## Database Setup Data model & Astra DB

# 04



## AI Train a text classifier

# 05



## API Bring to production

# 06



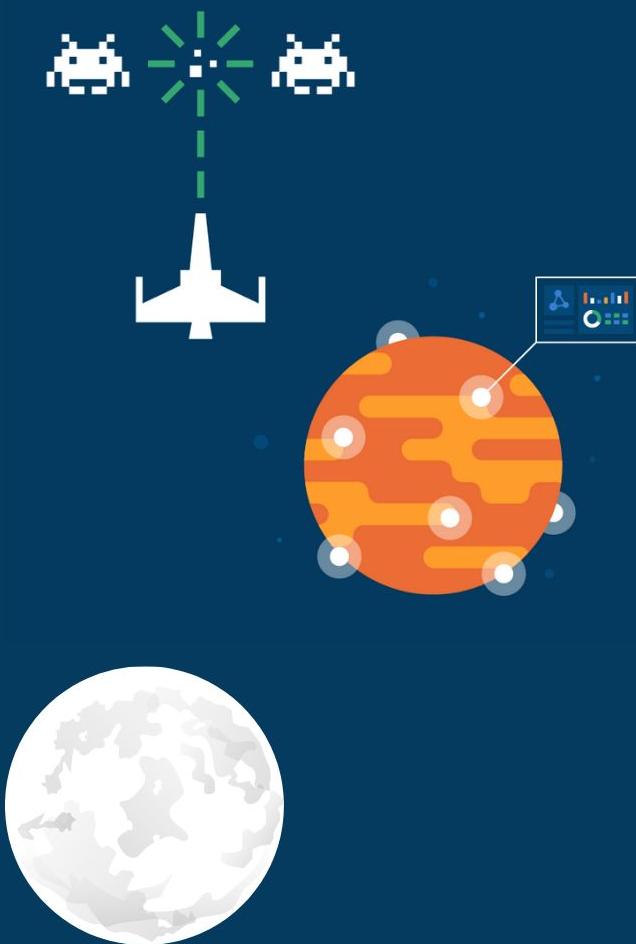
## What's next? Quiz, Homework, Agenda





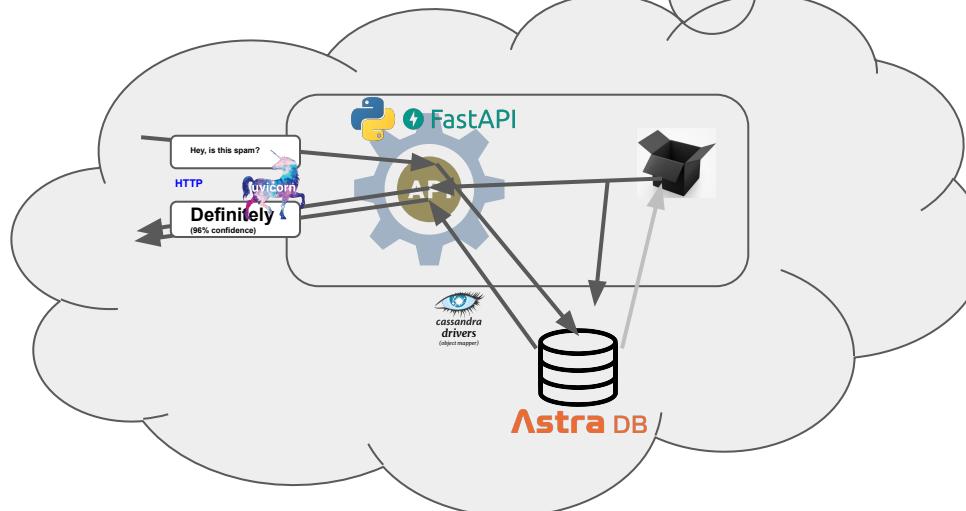
API

Bring to production





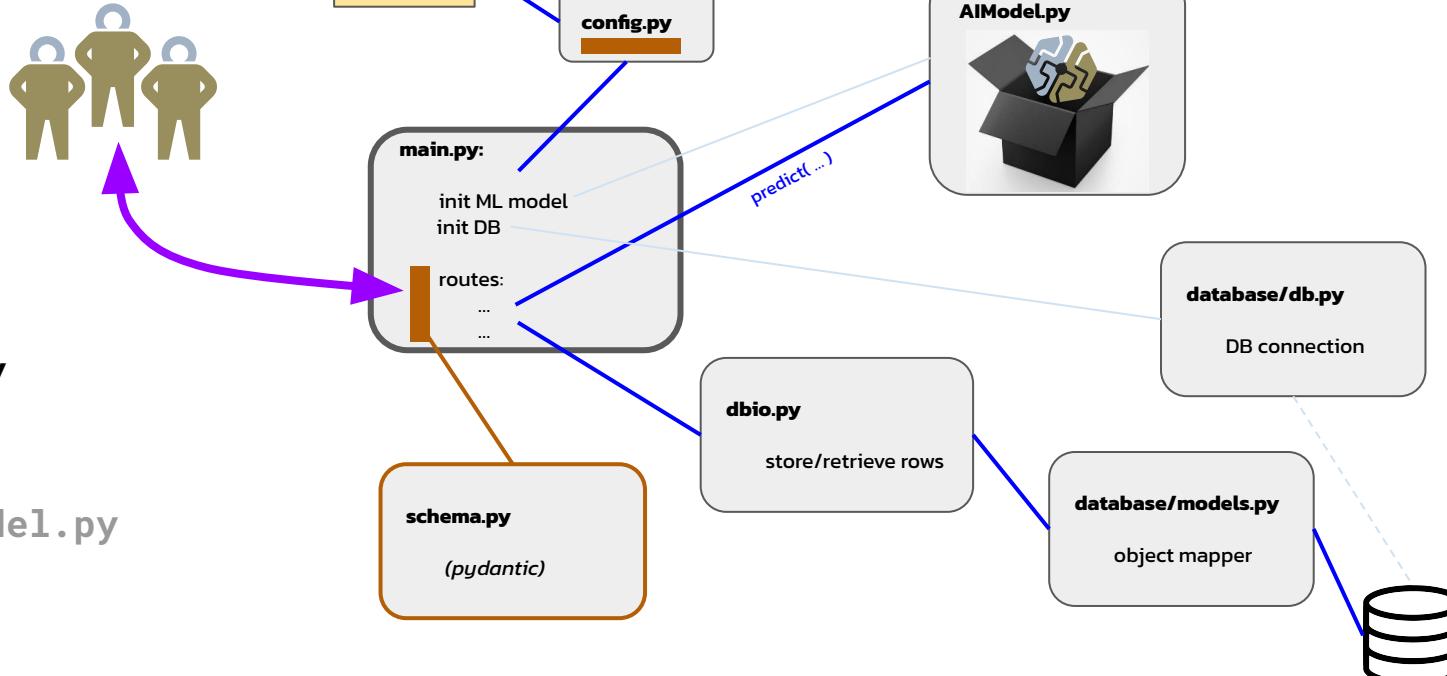
Here, take this



What now?

## api

```
└── main.py  
└── AIModel.py  
└── schema.py  
└── dbio.py  
└── database  
    └── db.py  
    └── models.py  
└── config.py  
└── minimain.py  
└── MockSpamAIModel.py  
└── tests (...)
```

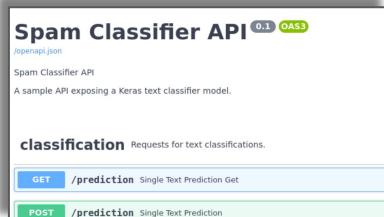


API structure

# ⚡ FastAPI fundamentals: routes, body validation

caching on DB to avoid re-processing the same input

API docs & Swagger UI



call logs & **StreamingResponse**

GET and query string

```
@app.get('/prediction', response_model=PredictionResult, tags=['classification'])
def single_text_prediction_get(request: Request, query: SingleTextQuery = Depends()):
```

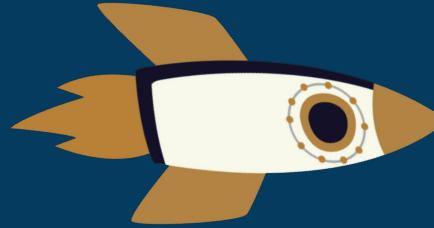
```
settings = getSettings()
cached = None if query.skip_cache else readCachedPrediction(settings.model_version)
storeCallsToLog([query.text], request.client[0])
if not cached:
    result = spamClassifier.predict([query.text], echoInput=query.echo_input)[0]
    cachePrediction(query.text, result)
    result['from_cache'] = False
    return PredictionResult(**result)
else:
    cached['from_cache'] = True
    return PredictionResult(**cached)
```

```
@app.get('/recent_log', response_model=List[CallerLogEntry], tags=['info'])
def get_recent_calls_log(request: Request):
    caller_id = request.client[0]
    called_hour = getThisHour()
    #
    return StreamingResponse(formatCallerLogJSON(caller_id, called_hour))
```



Features & highlights





# Hands-on (!github) API

- ✓ Expose the model with FastAPI
- ✓ Explore Swagger UI
- ✓ Look at the database



# 01

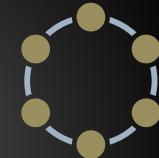


## Housekeeping Live and Hands-on



# 02

## Our Goal Use case and Technologies



# 03

## AI Train a text classifier



# 04

## API Bring to production



# 05

## What's next? Quiz, Homework, Agenda



# What's next?



Quiz, Homework, Agenda





DataStax Developers

# workshop-chat

<https://www.youtube.com/watch?v=MuwT5xxFVVI> - Subscribe to mailing list: [http...](http://)

Rechercher

PRESENTER — 1  
David Jones-Gilardi

HELPER — 7  
012345  
AaronP  
B1nary  
Chelsea Navo  
Jeremy Hanna  
John Sanda  
Patrick\_McFadin

EN LIGNE — 560  
-samu-  
6304-42JB  
Aahlya  
Abdurahim  
abhi3pathi  
Abhiis.s  
Abhineet  
Abirsh

Événements  
moderator-only  
. WELCOME  
start-here  
code-of-conduct  
introductions  
upcoming-events  
useful-resources  
memes  
your-ideas  
@the-stage

WORKSHOPS  
# workshop-chat  
# workshop-feedback  
workshop-materials  
upcoming-workshops

ASTRA DB  
getting-started  
astra-apis  
astra-development  
sample-applications

APACHE CASSANDRA  
Cedrick Lun...

RIGGITYREKT Hier à 21:14  
I have a 5 node datacenter, 4 nodes are on dse version 5.1.20, one is on dse5.0.15. I am doing some mixed version testing for a class and the one node that is 5.0.15 is coming up as an analytics workload. I dont have /etc/default/dse, instead I am using /etc/init.d/dse-cassandra.  
how do I make that node start in cassandra workload, not in analytics?

RIGGITYREKT Hier à 23:39  
Okay I found out my issue, when i started DSE 5.0.15 it had endpointsnitch set to DseSimpleSnitch, the rest of my cluster is using PropertyFileSnitch, when i change it to PropertyFileSnitch, it still uses the simple snitch config. looking at the docs i see there is a way to go to GossipingPropertyFileSnitch, but i need the property file one. I can wipe this dbs, do anything with this node to get this done. how do i fix this?  
@here

Erick Ramirez Aujourd'hui à 02:19  
mixed versions isn't supported and you're guaranteed to run into weird issues that will cause further problems down the track

Cedrick Lunen Aujourd'hui à 09:01  
When you start a node you have parameters -k for analytics, -g for graph and -s for search. To remove analytics check and remove -k

Envoyer un message dans #workshop-chat

# !discord

[dtsx.io/discord](https://dtsx.io/discord)



Datastax Developers Discord (18k+)



# Subscribe



# Subscribe



How to create an Authentication Token in...  
37 views • 4 weeks ago

How to use the Data Loader in Astra DB  
62 views • 4 weeks ago

Astra DB Sample App Gallery  
36 views • 4 weeks ago

How to use Secure Connect in Astra DB  
42 views • 4 weeks ago

Cassandra Day India: CL Room (Workshops)  
2.4K views • Streamed 4 weeks ago

Cassandra Day India: RF Room (Talks)  
1.3K views • Streamed 1 month ago



Badges

# Thank You!

