

CALLING ALL DEVELOPERS!

Introduction to machine Learning

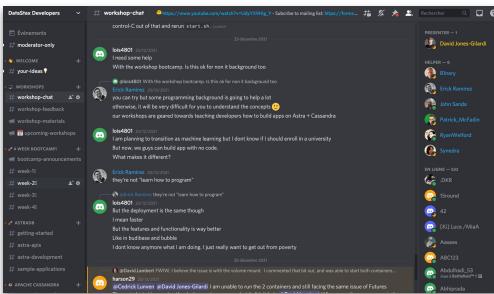
Apache Cassandra™
Apache Spark™

DataStax



Livestream: youtube.com/DataStaxDevs

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



YouTube
(with nightbot)

Discord
(#workshop-chat)

Games and quizzes: menti.com



Mentimeter

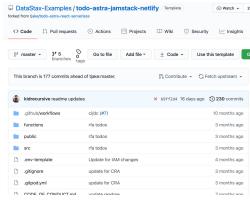


Live Sessions



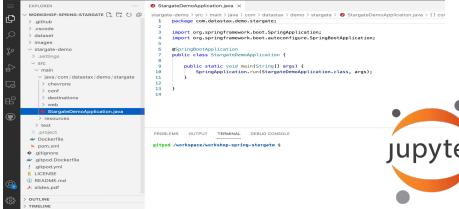
Nothing to install !

Source code + exercises + slides



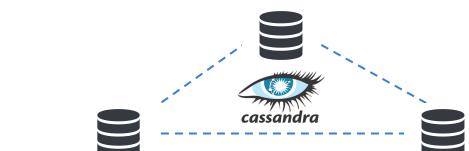
!github

IDE



!gitpod

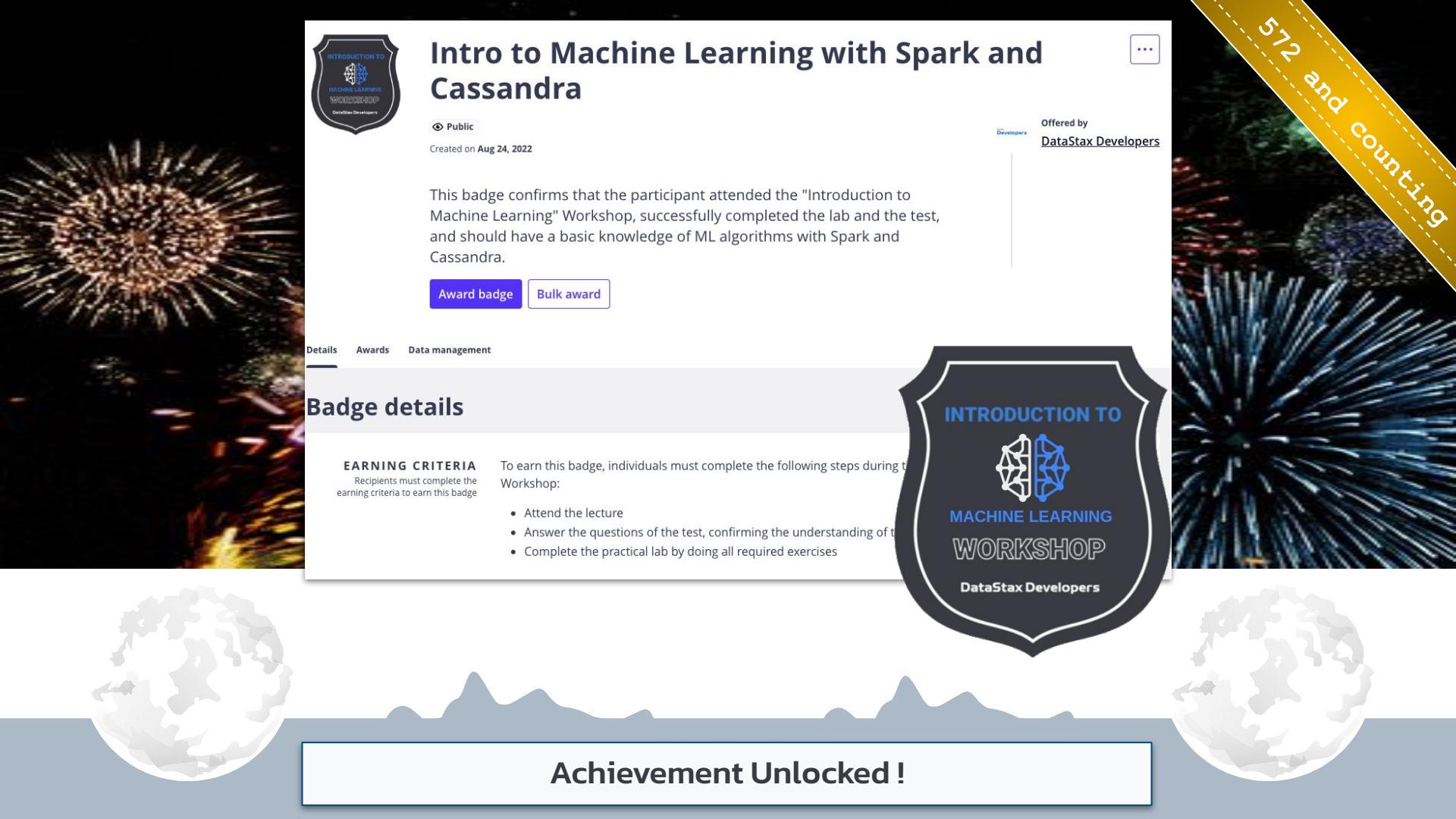
Database + Api + Streaming



DataStax
Astra DB

!astra

Hands-On Housekeeping





Intro to Machine Learning with Spark and Cassandra

Public

Created on Aug 24, 2022

This badge confirms that the participant attended the "Introduction to Machine Learning" Workshop, successfully completed the lab and the test, and should have a basic knowledge of ML algorithms with Spark and Cassandra.

[Award badge](#) [Bulk award](#)

[Details](#) [Awards](#) [Data management](#)

Badge details

EARNING CRITERIA
Recipients must complete the earning criteria to earn this badge

To earn this badge, individuals must complete the following steps during the Workshop:

- Attend the lecture
- Answer the questions of the test, confirming the understanding of the concepts
- Complete the practical lab by doing all required exercises



572 and counting

Achievement Unlocked !

01



Machine Learning
Definitions, Process, Metrics

02

Setup your env
Database, IDE, Tools

03

Learn about your tools
Cassandra, Jupyter, Spark

04

Algos 1
Clustering
Inference

05

Algos 2
Classification
Recommendation

06

What's next?
Quiz, Homework, Next week



Agenda

01



Machine Learning
Definitions, Process, Metrics

02

Setup your env
Database, IDE, Tools

03

Learn about your tools
Cassandra, Jupyter, Spark

04

Algos 1
Clustering
Inference

05

Algos 2
Classification
Recommendation

06

What's next?
Quiz, Homework, Next week



Agenda

Machine Learning



Machine learning is the scientific study of [algorithms](#) and [statistical models](#) that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

Wikipedia.org



What is Machine Learning ?



Machine learning is the scientific study of [algorithms](#) that can learn from and make predictions on data, typically without being explicitly programmed. In this sense, machine learning is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in spam filtering and computer vision, where it is difficult or unfeasible to develop a conventional algorithm to perform the needed task.

A. Volochnev

Wikipedia.org



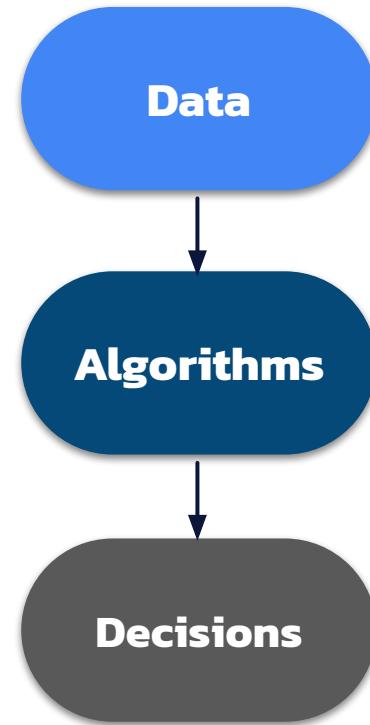
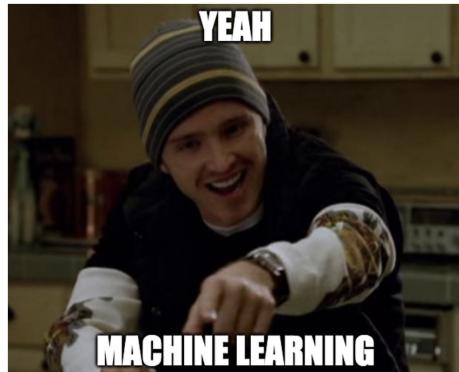
WIKIPEDIA
The Free Encyclopedia



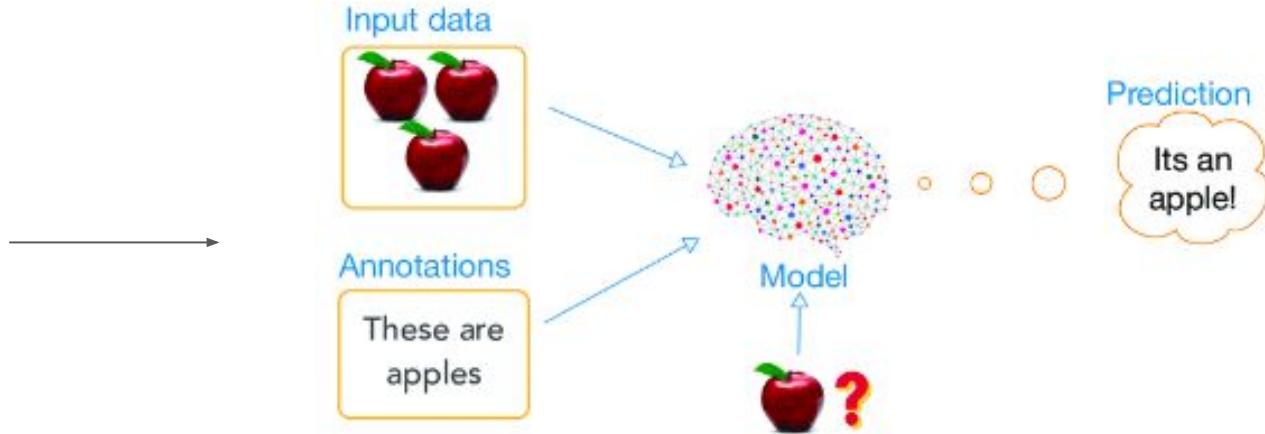
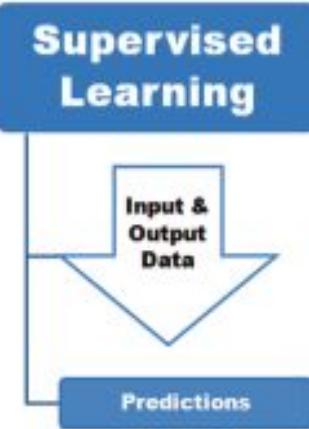
What is Machine Learning ?

Machine Learning is a scientific way to process raw data using algorithms to make better decisions.

No magic, just billions rows of data and two buckets of mathematics. Voilà!



How it works ?



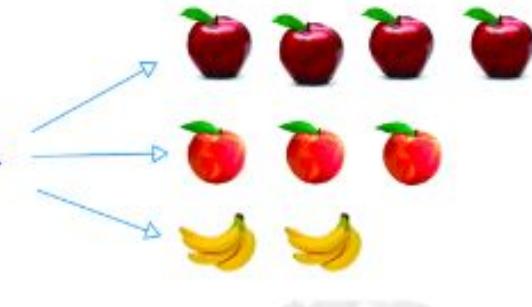
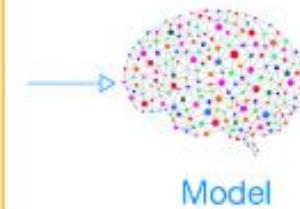
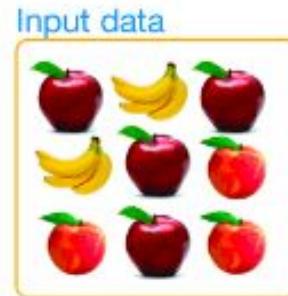
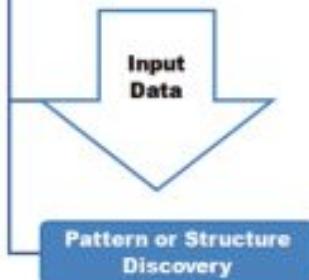
- **Knowledge of the output:** learn with expert
 - Data are **labelled** with class or value
 - **Goal:** predict the class



Supervised Learning



Unsupervised Learning



- **No Knowledge of the output:** self-guided
 - Data are not **labelled** with class or value
 - **Goal:** Determine Patterns of Grouping



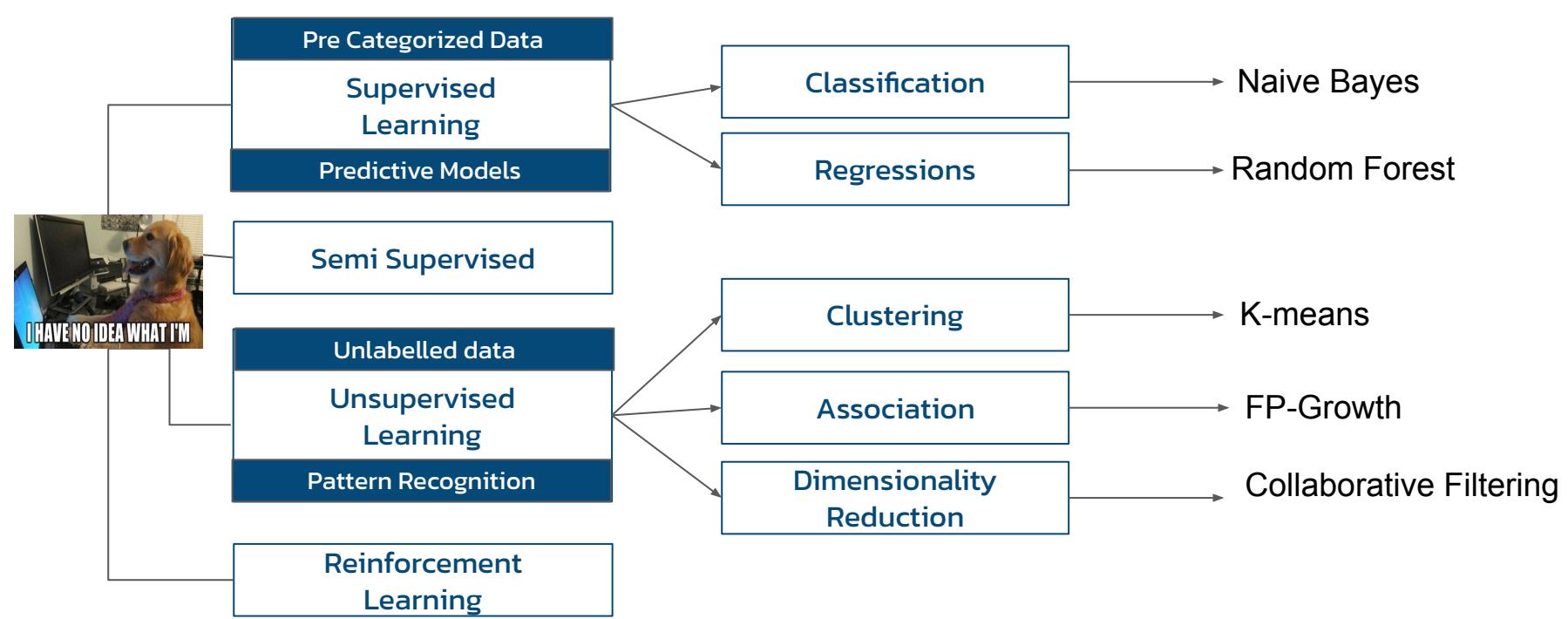
Unsupervised Learning

- FP-Growth
- K-Means Clustering
- Naive Bayes
- Decision Trees
- Neural Networks
- Collaborative Filtering
- Logistic Regression
- Support Vector Machines
- Linear Regression

- Apriori Algorithm
- Case-based Reasoning
- Dimensionality Reduction Algorithms
- Gradient Boosting Algorithms
- Hidden Markov Models
- Self-organizing Map
- K-Nearest Neighbour
- ECLAT

And still counting...





Data Model “Quality”



100 people, 9 have malignant tumor (very bad), 91 have benign tumor (bad)

True Positive (TP): <ul style="list-style-type: none">• Reality: Malignant• ML model predicted: Malignant• Number of TP results: 1	False Positive (FP): <ul style="list-style-type: none">• Reality: Benign• ML model predicted: Malignant• Number of FP results: 1
False Negative (FN): <ul style="list-style-type: none">• Reality: Malignant• ML model predicted: Benign• Number of FN results: 8	True Negative (TN): <ul style="list-style-type: none">• Reality: Benign• ML model predicted: Benign• Number of TN results: 90



Evaluating data model



Accuracy is an evaluating classification models metric, it is the fraction of predictions model identified correctly.

Correct Prediction (TP + TN)

$$\text{Accuracy} = \frac{\text{Correct Prediction (TP + TN)}}{\sum \text{ Predictions}}$$



Accuracy

True Positive (TP):

- Reality: Malignant
- ML model predicted: Malignant
- Number of TP results: 1

False Positive (FP):

- Reality: Benign
- ML model predicted: Malignant
- Number of FP results: 1

False Negative (FN):

- Reality: Malignant
- ML model predicted: Benign
- Number of FN results: 8

True Negative (TN):

- Reality: Benign
- ML model predicted: Benign
- Number of TN results: 90

What is the accuracy here ?



How many go home without proper treatment ?

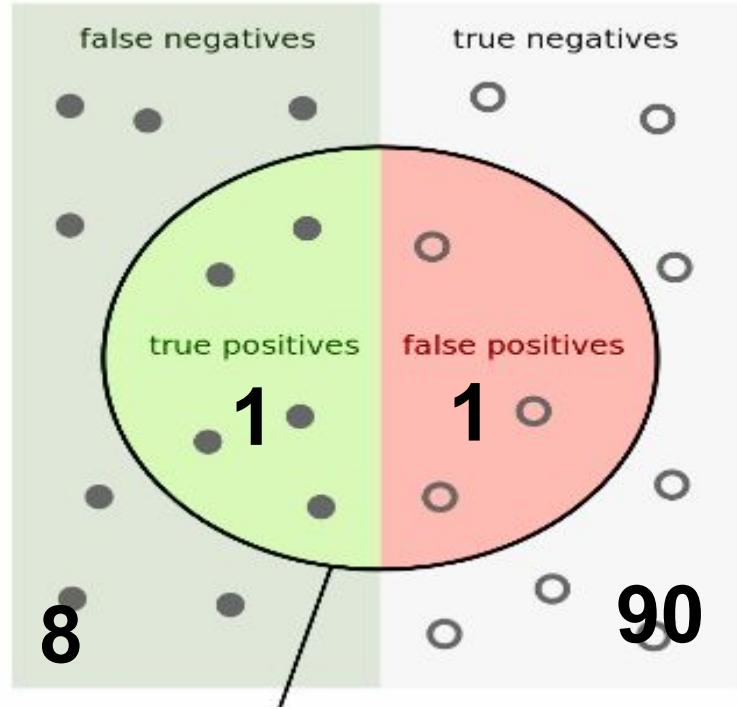


Precision counts true positives out of all true and false positives.

$$\text{Precision} = \frac{\text{True Positives (TP)}}{\sum \text{Positives (TP + FP)}}$$

What is the precision here?

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$



Precision

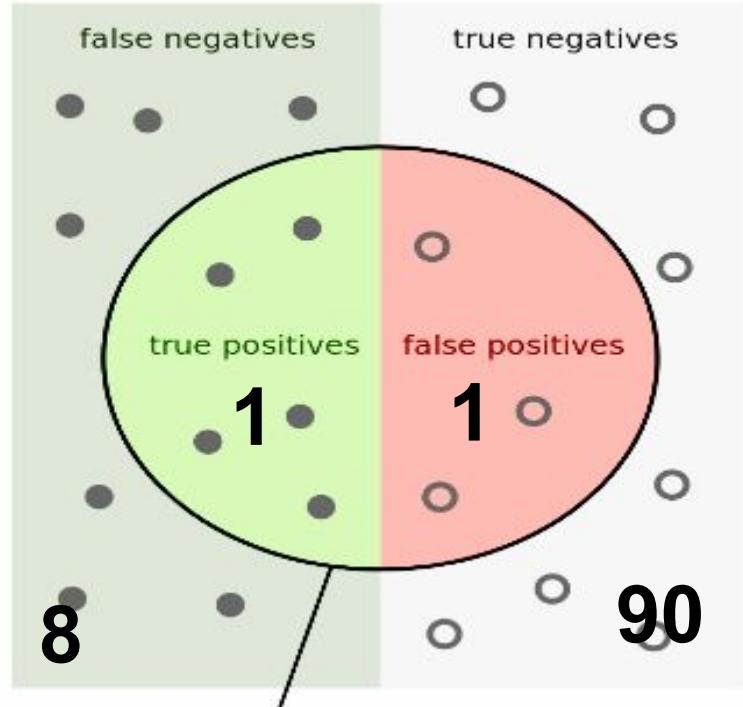


Recall correctly identified positives
out of all real positives.

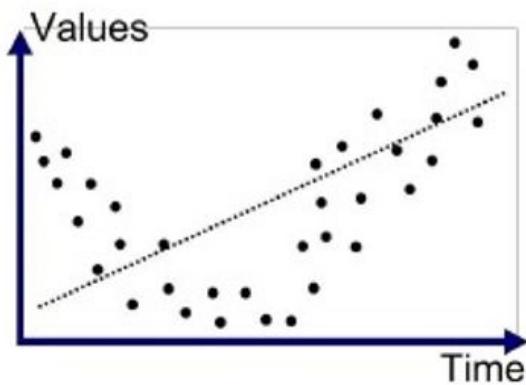
$$\text{Prediction} = \frac{\text{True Positives (TP)}}{\sum \text{Correct (TP + FN)}}$$

What is the recall here?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

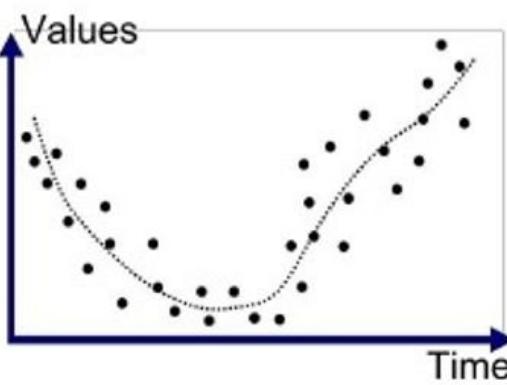


Recall



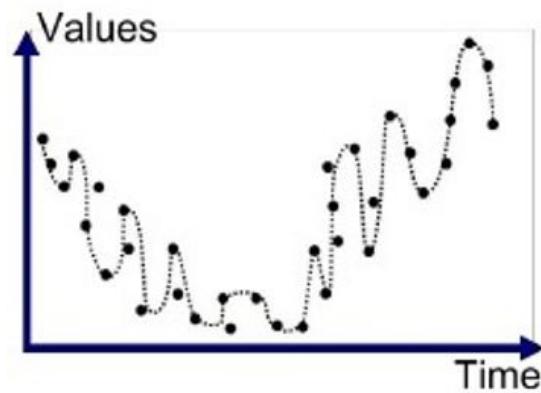
Underfitted

Not accurate, too simple



Good Fit/R robust

Good, well generalised



Overfitted

Over-trained, perfect on train data, fails on test data



Under fitted vs over-fitted model

01



Machine Learning
Definitions, Process, Metrics

02

Setup your env
Database, IDE, Tools

03

Learn about your tools
Cassandra, Jupyter, Spark

04

Algos 1
Clustering
Inference

05

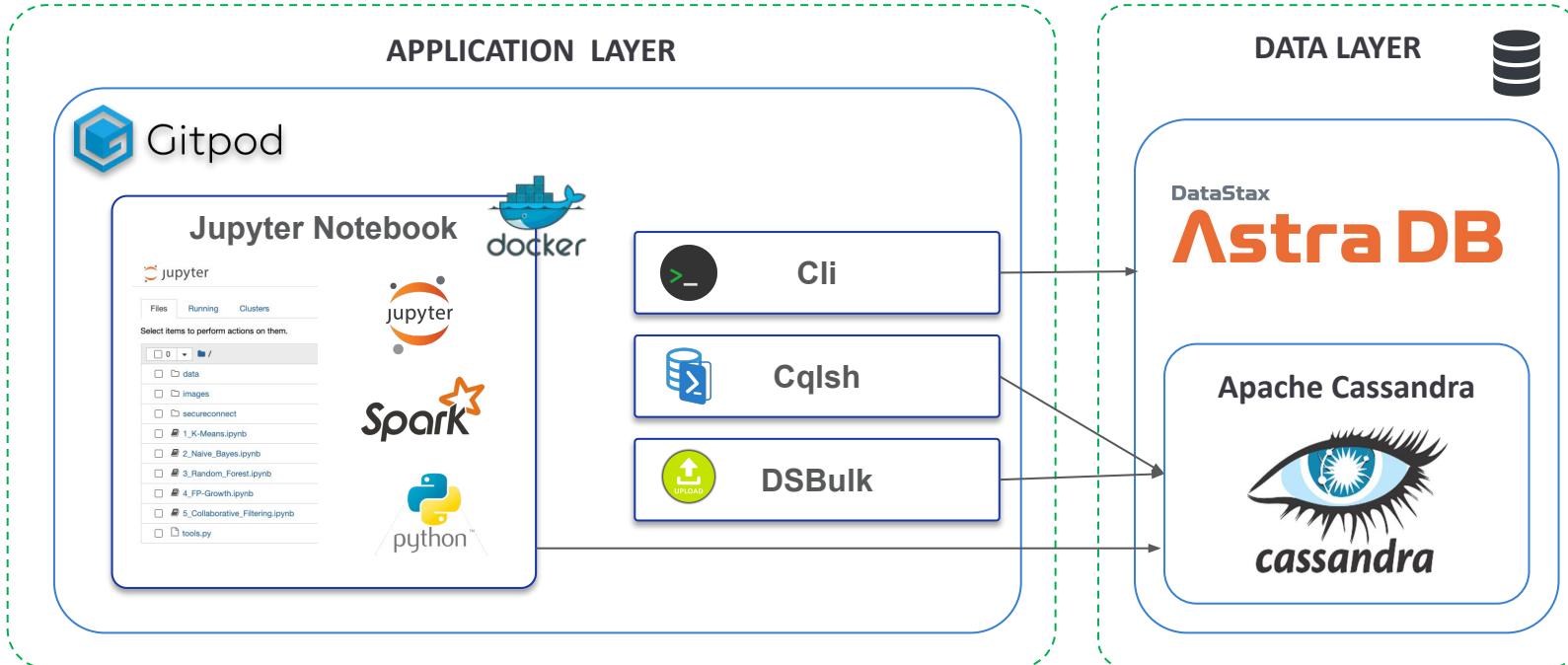
Algos 2
Classification
Recommendation

06

What's next?
Quiz, Homework, Next week



Agenda



Know your Tools

APPLICATION LAYER

DATA LAYER

DataStax

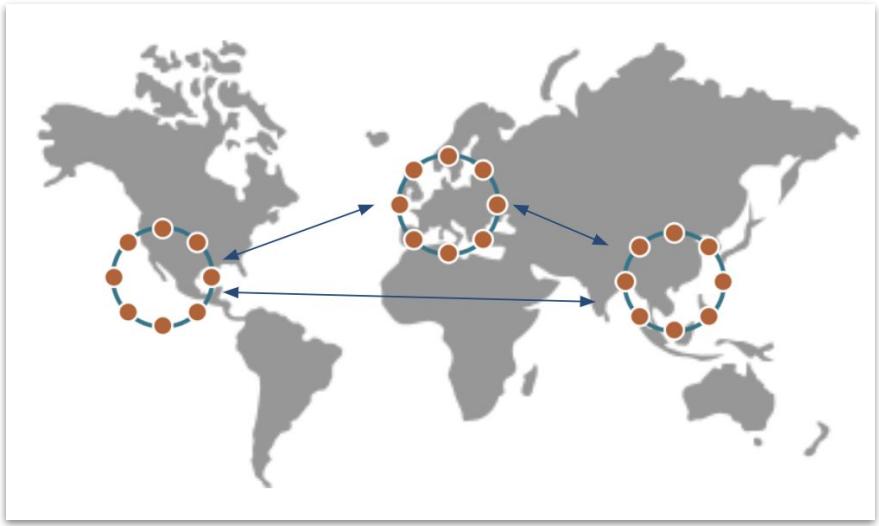
Astra DB

Apache Cassandra

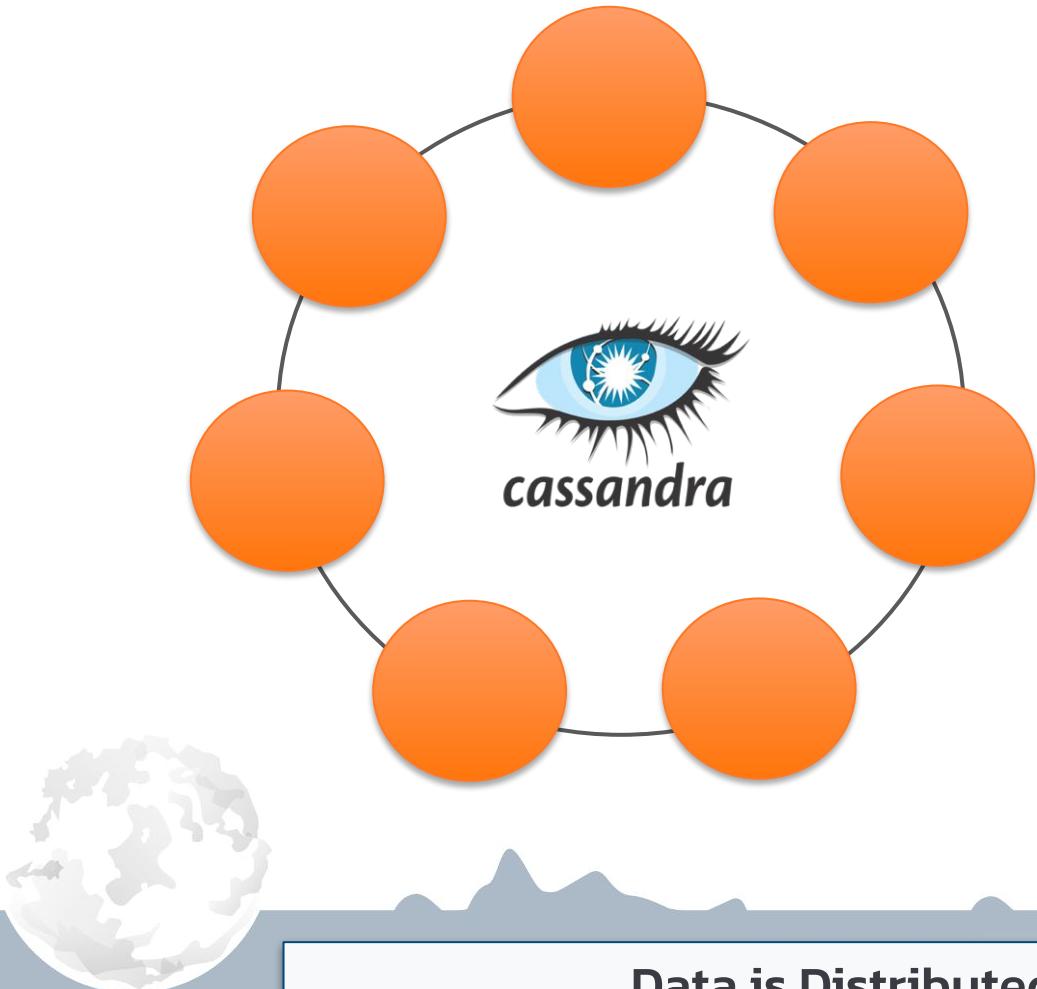


Know your Tools

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



Apache Cassandra



Country	City	Population
USA	New York	8.000.000
USA	Los Angeles	4.000.000
FR	Paris	2.230.000
DE	Berlin	3.350.000
UK	London	9.200.000
AU	Sydney	4.900.000
DE	Nuremberg	500.000
CA	Toronto	6.200.000
CA	Montreal	4.200.000
FR	Toulouse	1.100.000
JP	Tokyo	37.430.000
IN	Mumbai	20.200.000

Partition Key

Data is Distributed

USA	New York	8.000.000
USA	Los Angeles	4.000.000

Country City Population

DE	Berlin	3.350.000
DE	Nuremberg	500.000

FR	Paris	2.230.000
FR	Toulouse	1.100.000

UK	London	9.200.000
-----------	--------	-----------

JP	Tokyo	37.430.000
-----------	-------	------------

AU	Sydney	4.900.000
IN	Mumbai	20.200.000

CA	Toronto	6.200.000
CA	Montreal	4.200.000



cassandra



Data is Distributed

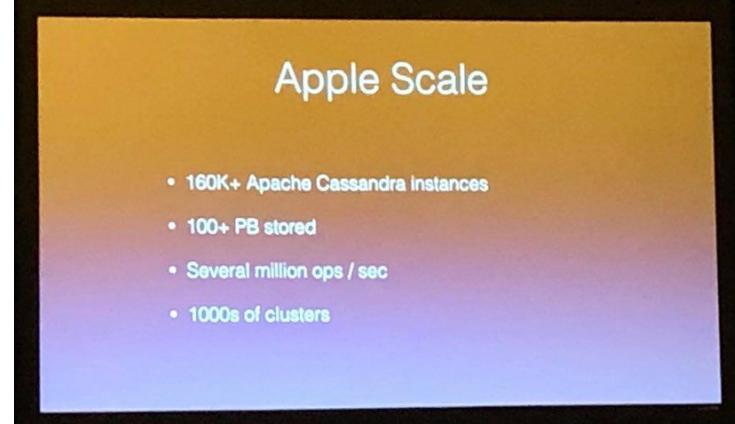
Apache Cassandra @ Netflix

- . 98% of streaming data is stored in Apache Cassandra
- . Data ranges from customer details to viewing history to billing and payments
- . Foundational datastore for serving millions of operations per second

- 30 million ops/sec on most active single cluster
- 500 TB most dense single cluster
- 9216 CPUs in biggest cluster

O(100) Clusters
O(10000) Instances
O(10,000,000) Replications per second
O(100,000,000) Operations per second
O(1,000,000,000,000,000) Petabytes of data

dtsx.io/cassandra-at-netflix



And many others...



Cassandra Biggest Users (and Developers)



Free to Use

Up to 80GB storage and/or 20 million operations monthly.



Serverless

Lower your costs by running Cassandra clusters only when needed.



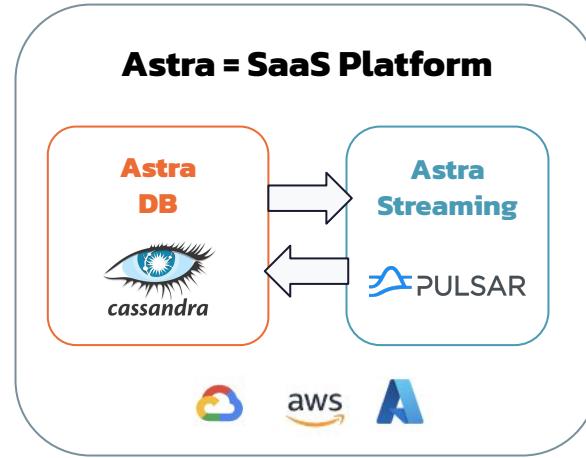
No Operations

Eliminate the overhead to install, operate, and scale Cassandra.



Data APIs

Work natively with Document (JSON), REST, GraphQL and gRPC APIs.



Global Scale

Put your data where you need it without compromising performance, availability or accessibility.



End-to-End Security

Secure connect with VPC peering and Private Link. Bring your own encryption key management. SAML SSO secure account access.



Zero Lock-in

Deploy on AWS, GCP or Azure and keep compatibility with open-source Cassandra.



Relational Indexes

Storage Attached Indexing (SAI) lets you query tables using any columns.



Astra = Cassandra As a Service++





Lab 1

Create Your Astra DB Instance

[https://github.com/datastaxdevs/workshop-introduction-to-machine-learning
#4-create-your-astra-db-instance](https://github.com/datastaxdevs/workshop-introduction-to-machine-learning#4-create-your-astra-db-instance)



01



Machine Learning
Definitions, Process, Metrics

02

Setup your env
Database, IDE, Tools

03

Learn about your tools
Cassandra, Jupyter, Spark

04

Algos 1
Clustering
Inference

05

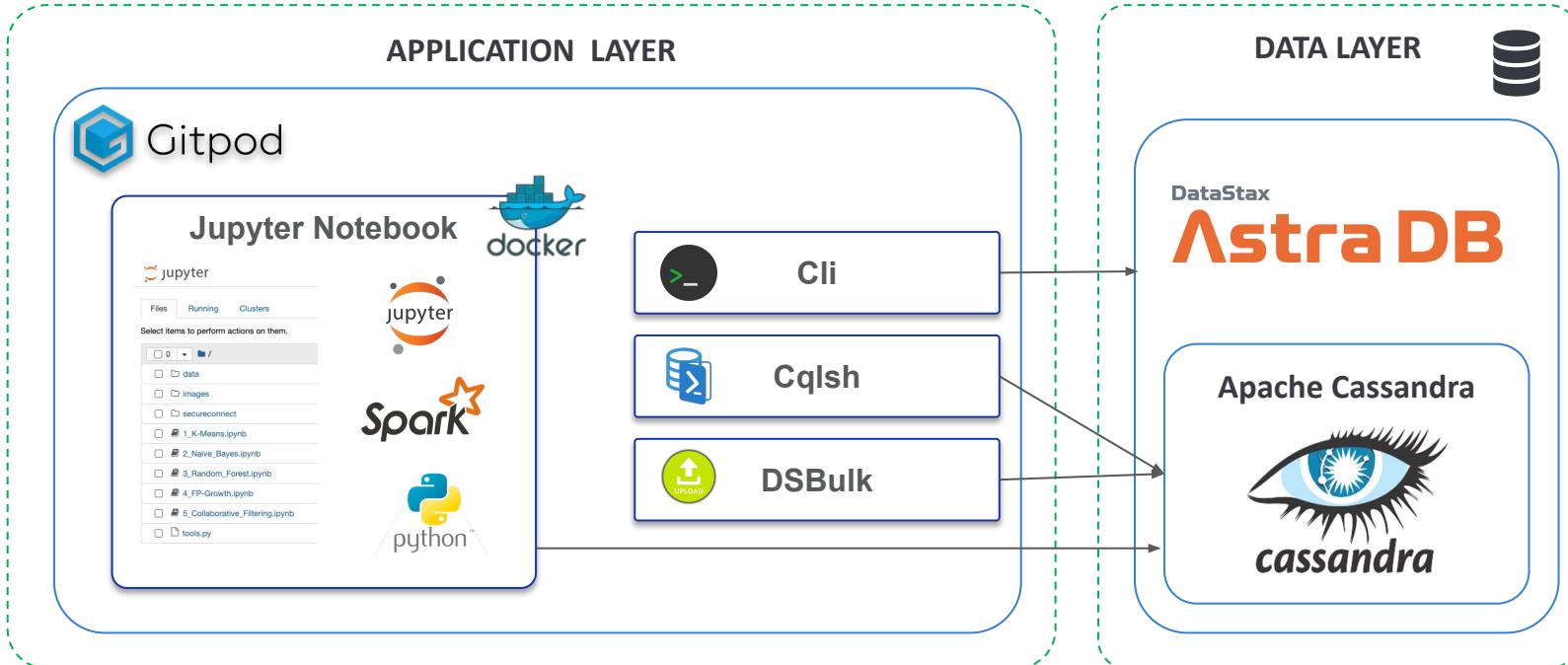
Algos 2
Classification
Recommendation

06

What's next?
Quiz, Homework, Next week



Agenda



Know your Tools

APPLICATION LAYER



Gitpod

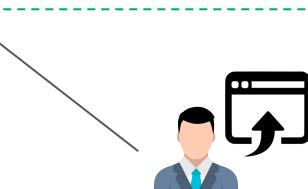
DATA LAYER



DataStax

Astra DB

Apache Cassandra



Gitpod

Always ready to code.

Spin up fresh, automated dev environments for each task, in the cloud, in seconds.

Try Now

Open a workspace.
Start from any Git context.



Star 8,829



```
version: '3'
services:
  jupyter: # Jupyter Notebook
    build: ./pyspark-cassandra
    volumes:
      - ./jupyter:/home/jovyan
    ports:
      - "8888:8888" # Exposes port to be available externally
environment:
  PYSPARK_SUBMIT_ARGS: '--packages com.datastax.spark:spark-cassandra-connector_2.11:2.5.1 pyspark-shell'
# see https://jupyter-docker-stacks.readthedocs.io/en/latest/using/common.html?highlight=start-notebook.sh#jupyter-server
# 'mlrules' (better to stick to sha1 to avoid escaping those pesky literal dollar signs, see https://stackoverflow.com/g
command: start-notebook.sh --NotebookApp.password='sha1:7b14dc0fd0:a01b354f5bb59e28352772811642bcc3dfb4b6'
# command: start-notebook.sh --NotebookApp.password='sha1:a536879cf56d:a895a85b375e09f7d6a8211cdcd0e87f16aa4e60'
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

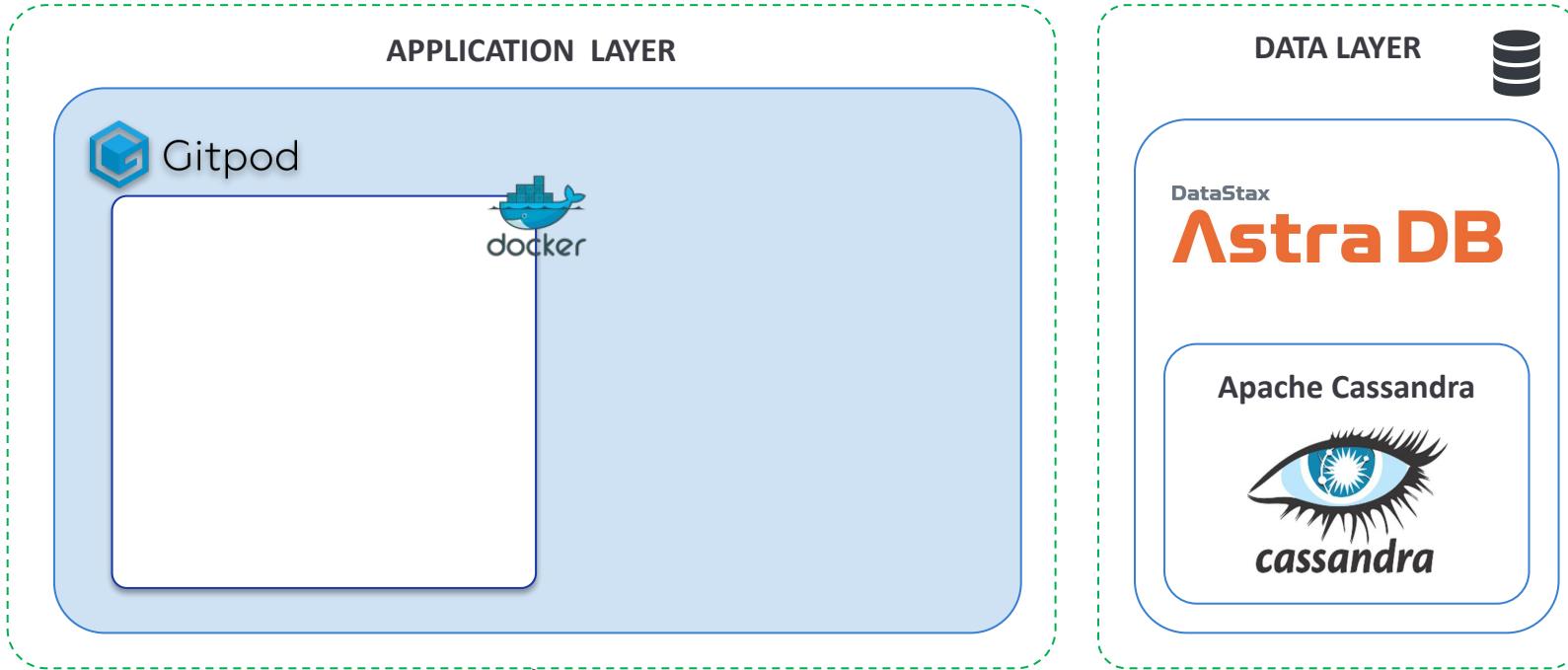
main-console: bash +v

** ASTRA DB CONFIGURATION **
Enter the required information to connect to your Astra DB instance
(make sure you have a DB token and a Secure Connect Bundle zipfile)
Enter Client ID from token :



Gitpod

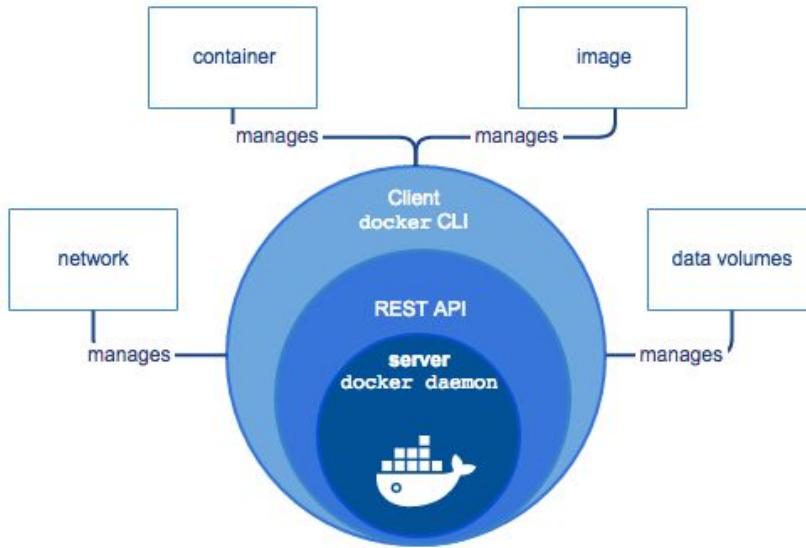




Gitpod



Docker Engine is Client-server application with :



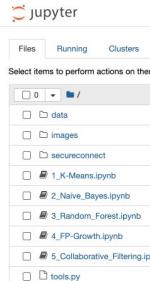
- A server which is a type of long-running program called a daemon process (the `dockerd` command).
- A REST API which specifies interfaces that programs can use to talk to the daemon and instruct it what to do.
- A command line interface (CLI) client (the `docker` command)



APPLICATION LAYER



Jupyter notebook



DATA LAYER



DataStax

Astra DB

Apache Cassandra



cassandra



Jupyter Notebook

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.



```
fileName = 'data/ratings.csv'  
input_file = open(fileName, 'r')  
  
for line in input_file:  
    row = line.split(',')  
  
    query = "INSERT INTO movieratings (userid, movieid, rating, timestamp)"  
    query += " VALUES (%s, %s, %s, %s)"  
    session.execute(query, (int(row[0]), int(row[1]), float(row[2]), row[3]))
```



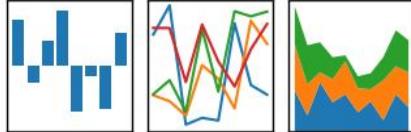
Python



Pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



```
In [1]: df = pd.DataFrame({'AAA': [4, 5, 6, 7],  
...: 'BBB': [10, 20, 30, 40],  
...: 'CCC': [100, 50, -30, -50]})  
...:
```

```
In [2]: df  
Out[2]:
```

	AAA	BBB	CCC
0	4	10	100
1	5	20	50
2	6	30	-30
3	7	40	-50



Apache Spark is written in Scala programming language. PySpark has been released in order to support the collaboration of Apache Spark and Python, it actually is a Python API for Spark. In addition, PySpark, helps you interface with Resilient Distributed Datasets (RDDs) in Apache Spark and Python programming language.



NumPy is the fundamental package for scientific computing with Python.

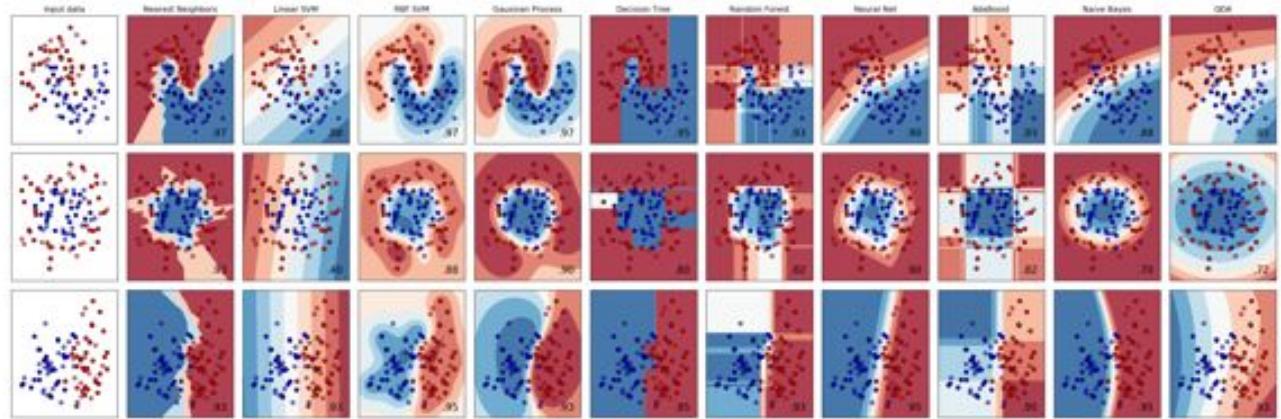


It contains among other things: a powerful N-dimensional array object, sophisticated functions, useful linear algebra, Fourier transform, and random number capabilities.

```
>>> x = np.array([('Rex', 9, 81.0), ('Fido', 3, 27.0)],  
...                 dtype=[('name', 'U10'), ('age', 'i4'), ('weight', 'f4')])  
>>> x  
array([('Rex', 9, 81.), ('Fido', 3, 27.)],  
      dtype=[('name', 'U10'), ('age', '<i4'), ('weight', '<f4')])
```



An open source, simple and efficient tool for predictive data analysis, accessible to everybody, and reusable in various contexts. Built on NumPy, SciPy, and matplotlib.



Scikit Lean



01



Machine Learning
Definitions, Process, Metrics

02

Setup your env
Database, IDE, Tools

03

Learn about your tools
Cassandra, Jupyter, Spark

04

Algos 1
Clustering
Inference

05

Algos 2
Classification
Recommendation

06

What's next?
Quiz, Homework, Next week



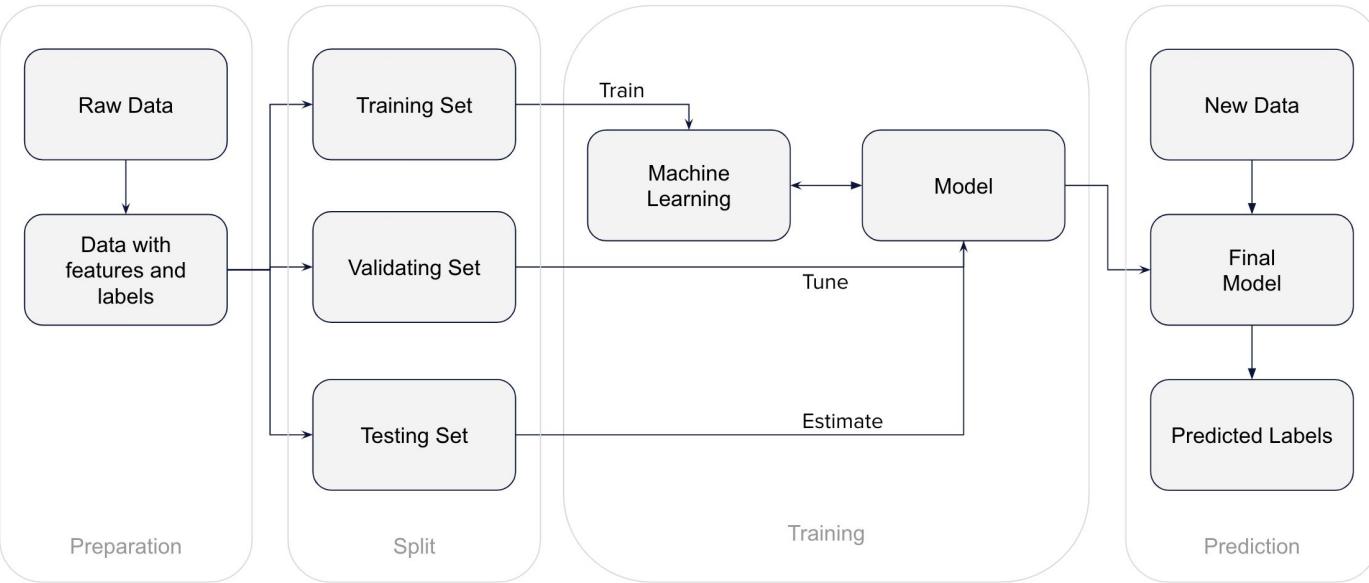
Agenda

Demo with K-means

<http://user.ceng.metu.edu.tr/~akifakkus/courses/ceng574/k-means/>



- **Question / Hypothesis**
- **Algorithm Selection**
- **Data Preparation**
- **Data Split**
- **Training**
- **Tuning**
- **Testing**
- **Analysis**
- **Repeat**



Learning Workflow

Raw Data

Training Set

Train

Machine Learning

Model

Data with
features and
labels

Validating Set

Tune

Testing Set

Estimate

New Data

Final Model

Predicted Labels

Preparation

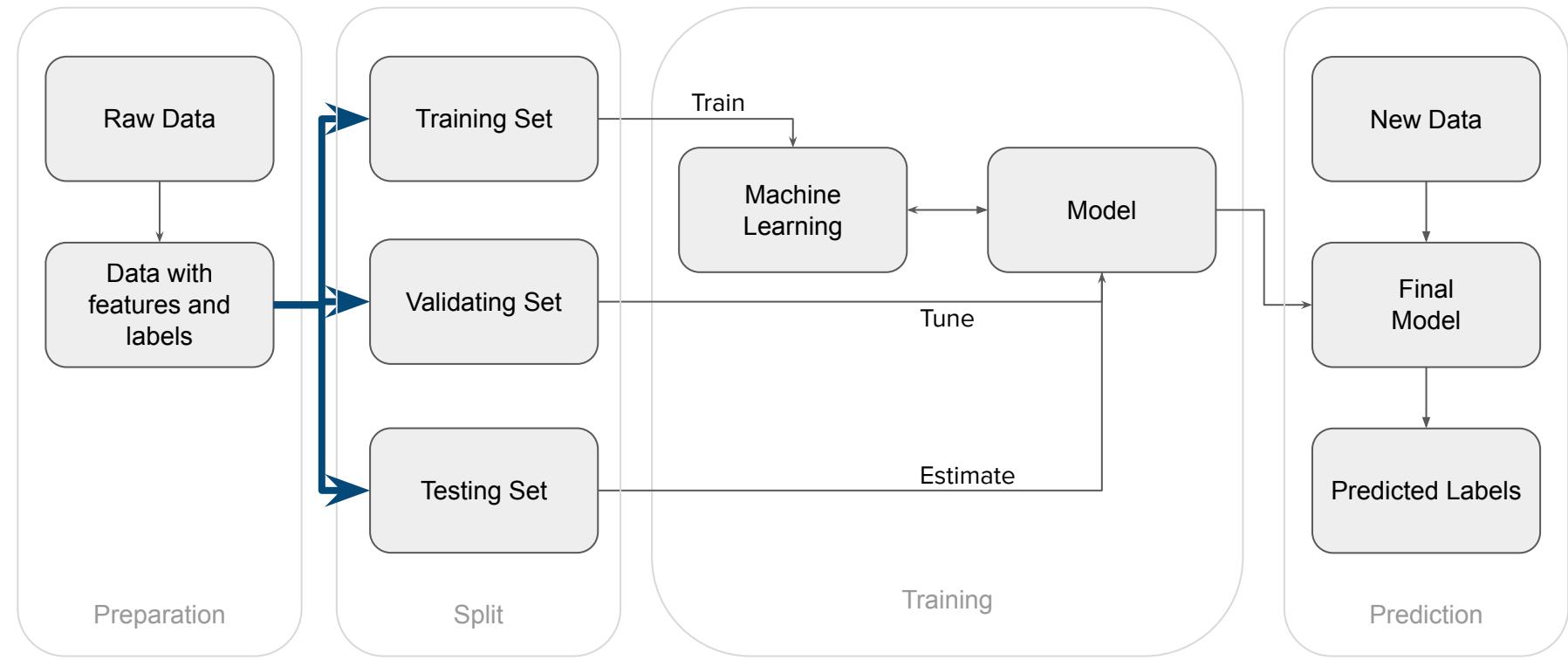
Split

Training

Prediction



Data Preparation



Data Split



Raw Data

Data with
features and
labels

Training Set

Validating Set

Testing Set

Train

Machine
Learning

Model

Tune

Estimate

New Data

Final
Model

Predicted Labels

Preparation

Split

Training

Prediction



Training





FUN FACT: this image was created by ... an algorithm,
starting from the textual prompt: "**a metallic cyborg in a gym**"

<https://huggingface.co/spaces/stabilityai/stable-diffusion>

Intermezzo: "training the machine"

Raw Data

Data with
features and
labels

Training Set

Validating Set

Testing Set

Train

Machine
Learning

Model

Tune

Estimate

New Data

Final
Model

Predicted Labels

Preparation

Split

Training

Prediction



Tuning



Raw Data

Data with
features and
labels

Training Set

Validating Set

Testing Set

Model

Train

Machine
Learning

Tune

Estimate

New Data

Final
Model

Predicted Labels

Preparation

Split

Training

Prediction



Testing



Raw Data

Data with
features and
labels

Training Set

Validating Set

Testing Set

Train

Machine
Learning

Model

Tune

Estimate

New Data

Final
Model

Predicted Labels

Preparation

Split

Training

Prediction



Final Model

Raw Data

Data with
features and
labels

Training Set

Validating Set

Testing Set

Train

Machine
Learning

Model

Tune

Estimate

New Data

Final
Model

Predicted Labels

Preparation

Split

Training

Prediction



Prediction



Lab 2

K Means

[https://github.com/databricks/workshop-introduction-to-machine-learning
#6-algorithms](https://github.com/databricks/workshop-introduction-to-machine-learning#6-algorithms)



Naïve Bayes

Supervised Classification Algorithm



"Naïve Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong "naïve" independence assumptions between the features.

Naïve Bayes is a popular method for text categorization, the problem of judging documents as belonging to one category or the other (such as spam or legitimate, etc.) with word frequencies as the features. With appropriate pre-processing, it is competitive in this domain with more advanced methods including support vector machines. It also finds application in automatic medical diagnosis."

$$p(D | S) = \prod_i p(w_i | S)$$

and

$$p(D | \neg S) = \prod_i p(w_i | \neg S)$$

Using the Bayesian result above, we can write:

$$p(S | D) = \frac{p(S)}{p(D)} \prod_i p(w_i | S)$$

$$p(\neg S | D) = \frac{p(\neg S)}{p(D)} \prod_i p(w_i | \neg S)$$

Dividing one by the other gives:

$$\frac{p(S | D)}{p(\neg S | D)} = \frac{p(S) \prod_i p(w_i | S)}{p(\neg S) \prod_i p(w_i | \neg S)}$$

Which can be re-factored as:

$$\frac{p(S | D)}{p(\neg S | D)} = \frac{p(S)}{p(\neg S)} \prod_i \frac{p(w_i | S)}{p(w_i | \neg S)}$$

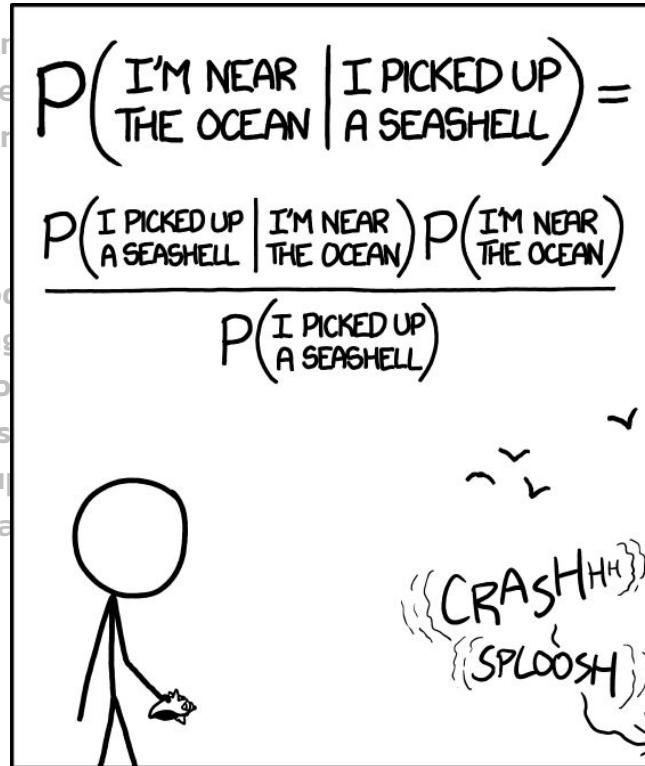


Naïve Bayes Algorithm



"Naïve Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naïve) assumptions between the features."

Naïve Bayes is a popular method for classifying documents (e.g., spam or legitimate, etc.) with words. After appropriate pre-processing, it is often competitive with more advanced methods including support vector machines. It has found application in automatic medical diagnosis.



STATISTICALLY SPEAKING, IF YOU PICK UP A SEASHELL AND DON'T HOLD IT TO YOUR EAR, YOU CAN PROBABLY HEAR THE OCEAN.

Naïve Bayes Algorithm

Naïve Bayes Algorithm

For Humans

Step I: Prepare and label data

- Get cash easy!
- Win money now!
- Greetings from Dad
- Could you lend me money?
- It was easy!
- I miss you
- Check your trip itinerary



Spam:

- Money transfer received
- Get cash easy!
- Win money now!

Not Spam:

- Greetings from Dad
- Could you lend me money?
- It was easy!
- I miss you
- Check your trip itinerary

Naïve Bayes Algorithm

For Humans

Spam:

- Money transfer received
- Get cash easy!
- Win money now!

Not Spam:

- Greetings from Dad
- Could you lend me money?
- It was easy!
- I miss you
- Check your trip itinerary

58

New Email: “Easy Money!” Is it spam?

Human: What is the chance it's spam if it has ‘Easy Money’?

Scientist: $P(\text{spam} \mid \text{'Easy', 'Money'}) = ?$

Money: 2 out of 3 spam $\Rightarrow P(\text{'Money'} \mid \text{spam}) = \frac{2}{3}$

1 out of 5 not $\Rightarrow P(\text{'Money'} \mid \text{not}) = \frac{1}{5}$

Easy: 1 out of 3 spam $\Rightarrow P(\text{'Easy'} \mid \text{spam}) = \frac{1}{3}$

1 out of 5 not $\Rightarrow P(\text{'Easy'} \mid \text{not}) = \frac{1}{5}$

Conditional Probability! $P(\text{spam}) + P(\text{not}) = \frac{3}{8} + \frac{5}{8} = 1.0$

$P(\text{spam} \mid \text{'Easy', 'Money'}) \propto P(\text{'Easy', 'Money'} \mid \text{spam}) * P(\text{spam})$

$\propto P(\text{'Easy'} \mid \text{spam}) * P(\text{'Money'} \mid \text{spam}) * P(\text{spam})$

$\frac{1}{3}$

$\frac{2}{3}$

$\frac{3}{8}$

$P(\text{not} \mid \text{'Easy', 'Money'}) \propto P(\text{'Easy', 'Money'} \mid \text{not}) * P(\text{not})$

$\propto P(\text{'Easy'} \mid \text{not}) * P(\text{'Money'} \mid \text{not}) * P(\text{not})$

$\frac{1}{5}$

$\frac{1}{5}$

$\frac{5}{8}$

$P(\text{spam} \mid \text{'Easy', 'Money'}) \propto \frac{1}{3} * \frac{2}{3} * \frac{3}{8} \propto 1 / 12 \Rightarrow p = 10/13$

$P(\text{not} \mid \text{'Easy', 'Money'}) \propto \frac{1}{5} * \frac{1}{5} * \frac{5}{8} \propto 1 / 40 \Rightarrow p = 3/13$

$P(\text{spam} \mid \text{'Easy', 'Money'}) = 0.7692307 = 76.9\%$

Okey, but why Naïve? Because it doesn't consider relations between facts. For example, if I write a word "Happy", the probability of the next word to be "Birthday" is obviously higher than "Funerals". Say "long long ago" and a person next to you most probably will continue: "in a galaxy far far away". This algorithm is called Naïve because it considers every fact as a stand-alone, not related to others.

It seems to be a serious flaw but surprisingly it isn't – on the reasonable amounts of data the NB may outperform many other more sophisticated algorithms. Additionally, it's a great for the parallel computing which makes it lightning fast.

Speaking to Bayes:

- Is it warm outside?
- Yes.
- Is it cold outside?
- No.

Speaking to a Human:

- Is it warm outside?
- Yes.
- Is it cold outside?
- Are you an idiot!?



Naïve Bayes Algorithm





Lab 3

Naive Bayes

[https://github.com/datastaxdevs/workshop-introduction-to-machine-learning
#6-algorithms](https://github.com/datastaxdevs/workshop-introduction-to-machine-learning#6-algorithms)



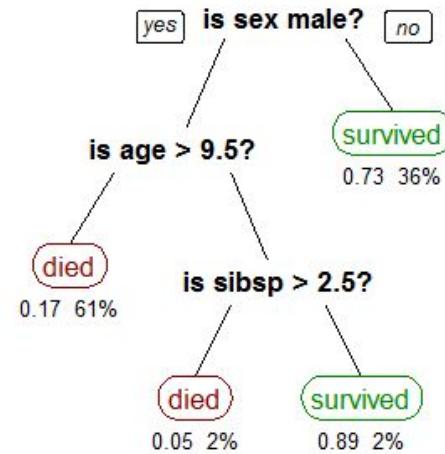
Random Forest

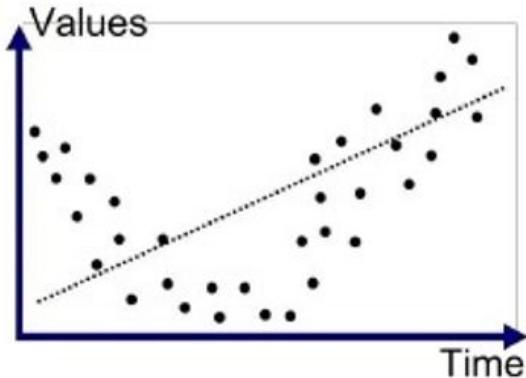
Supervised Classification Ensemble Method



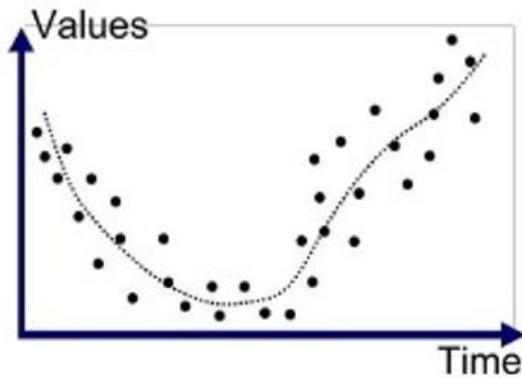
"A decision tree is a simple representation for classifying. Assume that all of the input features have finite discrete domains, and there is a single target feature called the "classification". Each element of the domain of the classification is called a class. A decision tree is a tree in which each internal node is labeled with an input feature. The arcs coming from a node labeled with an input feature are labeled with each of the possible values of the target or output feature or the arc leads to a subordinate decision node on a different input feature. Each leaf of the tree is labeled with a class or a probability distribution over the classes, signifying that the data set has been classified by the tree into either a specific class, or into a particular probability distribution"

Titanic Survival Decision Tree

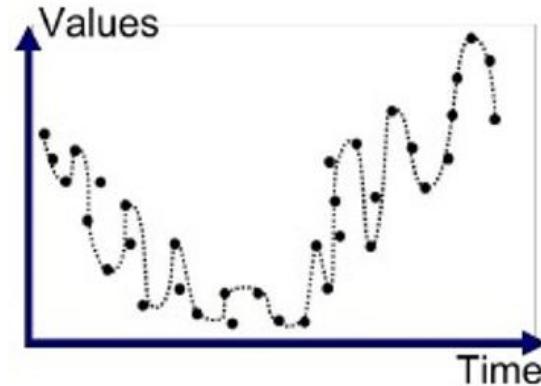




Underfitted



Good Fit/Robust



Overfitted



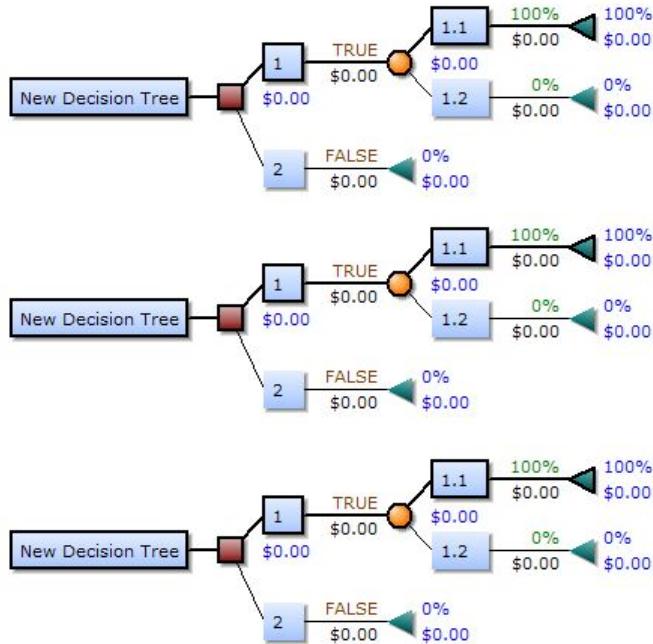
Under-fitter vs over-fitted



Random Forest Ensemble Method

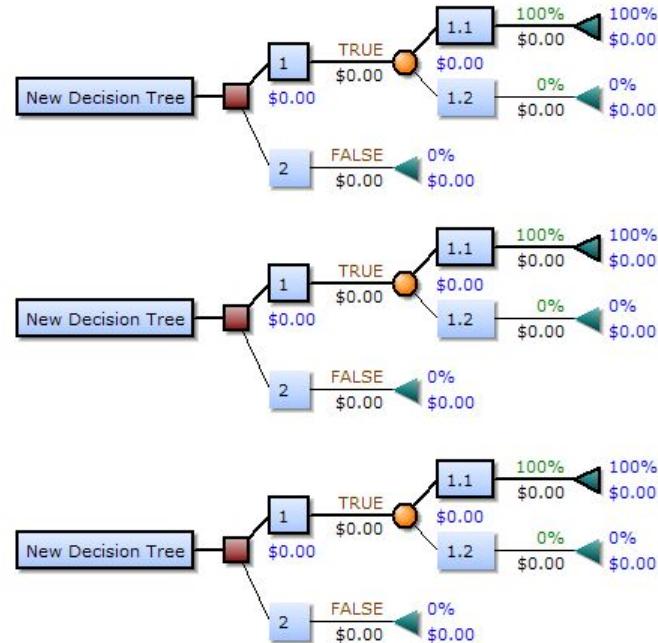
“Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

The first algorithm for random decision forests was created by Tin Kam Ho using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg.”



"Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set."

The first algorithm for random decision forests was created by Tin Kam Ho using the random subspace method, which, in Ho's formulation, is a way to implement the "stochastic discrimination" approach to classification proposed by Eugene Kleinberg."



Random Forest Ensemble





Lab 4

Random Forest

[https://github.com/databricks/workshop-introduction-to-machine-learning
#6-algorithms](https://github.com/databricks/workshop-introduction-to-machine-learning#6-algorithms)



FP-Growth

Association-rule-based recommendation system



Association-rule learning

A class of models to infer relations between elements in a list of sets. E.g. market-basket analysis:

"Customers who bought x, y, z also bought w"

Generally proceed in two steps:

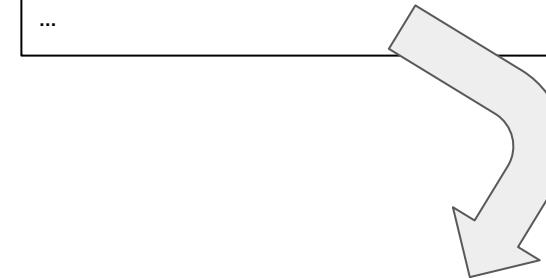
1. identify items that are often together
2. recast this information as "association rules"

FP-Growth

A method to build the "frequent itemsets" in 2 passes

Start from item counts and then rearrange items in a tree (a *trie*): frequent itemsets are found as paths in this tree.

user1: bread, tomatoes, mayonnaise
user2: ham, cheese, mayonnaise
user3: bread, ham, mayonnaise, pickles
user4: beer, diapers :)
user5: beer, ham, milk
...



$\{bread, ham\} \Rightarrow \{mayonnaise\}$



FP-Growth



Collaborative Filtering

Matrix-factorization-based recommendation system



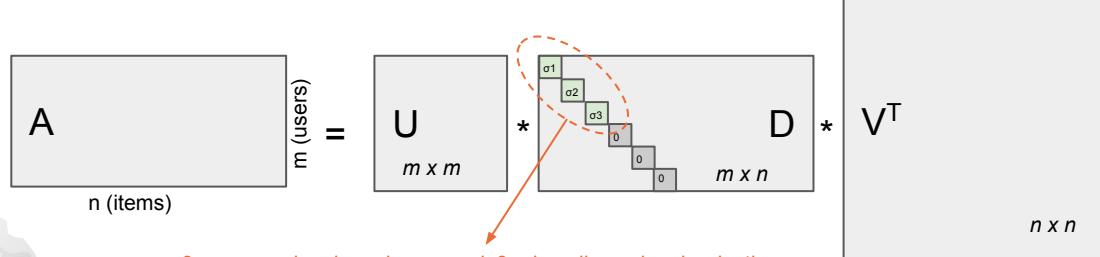
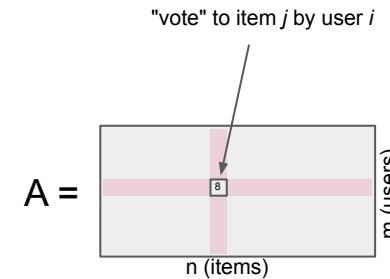
Users "collaborate" in identifying your tastes

Idea: "If you share the wine and snack tastes of a group of users, you will also like the same meals"

More precisely (model-based filtering):

- arrange preferences in a matrix A (users x items)
- Recast the matrix (*singular value decomposition*)
- Highlight the low-dimensional *latent space*

This finds the "core groups and their core tastes", an optimized form of (most of) the full information



Collaborative Filtering



01



Machine Learning
Definitions, Process, Metrics

02

Setup your env
Database, IDE, Tools

03

Learn about your tools
Cassandra, Jupyter, Spark

04

Algos 1
Clustering
Inference

05

Algos 2
Classification
Recommendation

06

What's next?
Quiz, Homework, Next week



Agenda

Homework

! homework

To gain your Badge:

- answer some "theory" questions
- do the assignment: improve the accuracy of the Random Forest classifier. **Details in the README.**



Join our 17k Discord Community

DataStax Developers



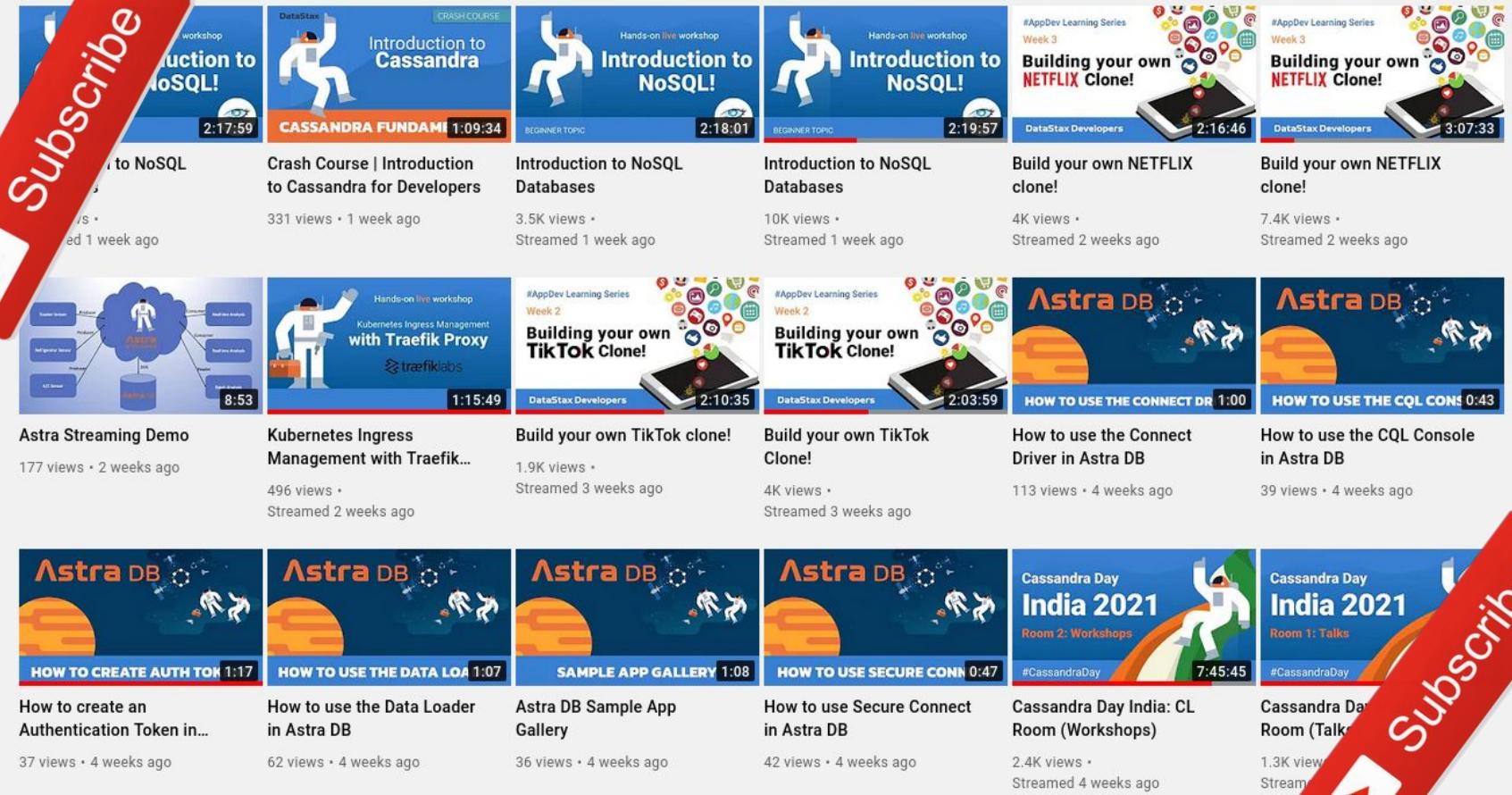
!discord

dtsx.io/discord

The screenshot shows the DataStax Developers Discord server interface. The left sidebar lists various channels: 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, # ASTRADB, # getting-started, # astra-apis, # astra-development, # sample-applications, and # APACHE CASSANDRA. The main window is the # workshop-chat channel, which contains a message from a user named RIGGITYREKT asking about node start configurations. Another message from Cedrick Lunven provides a solution. On the right side, there are lists for PRESENTER — 1 (David Jones-Giardi), HELPER — 7 (012345, AaronP, Binary, Chelsea Navo, Jeremy Hanna, John Sanda, Patrick_McFadin), and EN LIGNE — 560 (-samu-, 6304-42JB, Aahlya, Abdurahim, abhi3pathi, Abhiis.s, Abhineet, Abirish). At the bottom, there's a message input field: Envoyer un message dans #workshop-chat.

Discord Community

Subscribe



Subscribe

DataStax Developers

Thank You!

