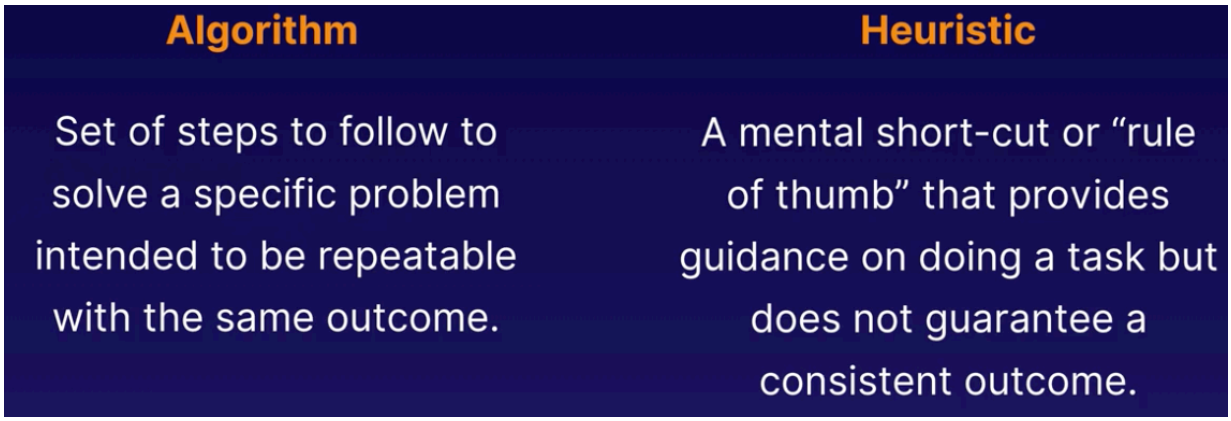
Cloud Guru - 6 - Algorithms

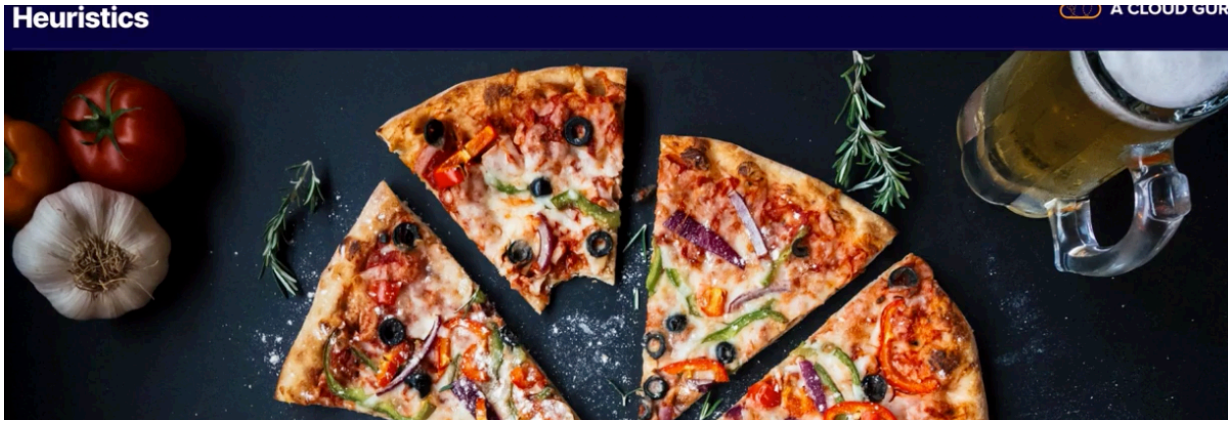
https://acloud.guru/course/aws-certified-machine-learning-specialty/learn/1efec92d-271f-f097-0c26-236b41ab42b1/f4cd49ae-be82-06a0-7672-7edc64126b7f/watch?backUrl=~2Fcourses&backUrl=~2Fcourses&backUrl=~2Fcourses,~2Fcourses

Algorithm vs Heuristic



Algorithm = troubleshooting steps to narrow down possibilities to the correct one
Ex: for electrical pb - go through a set of steps to see where the fault is

Heuristic = educating guess
Ex:We try a new pizza place. It could be a good or bad pizza, but we don't know. We are not going to try to see how they make the tough, what is their recipe, match against what we know,... We are willing to take a chance here
We use heuristic a lot in our life



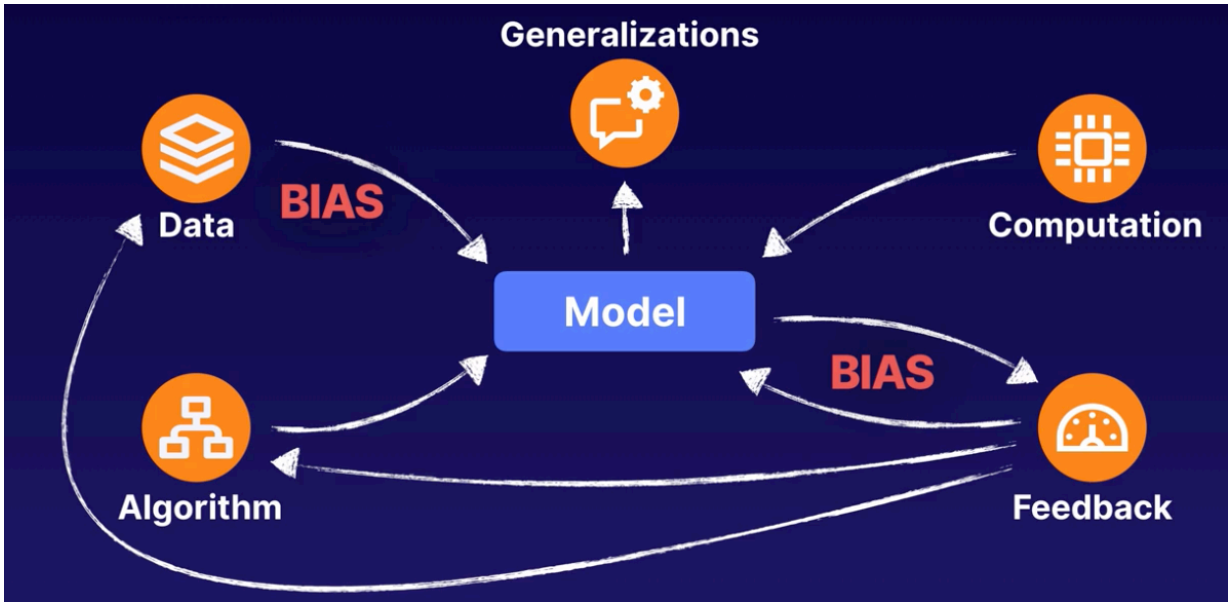
When to use algorithm - trying to **drive out Bias**
Trying to remove assumptions and past experience.

In contrast, computers can't make these assumptions. They don't have that intuition required by a heuristic.

=> **Algorithm** is a **finite process with known input and expected outputs**
We can use this in ML as we want repeatable generalization.

Now **Bias** can still creep in into ML process:
- by data we select when training or testing, or when we exclude a chunk of data
- feedback loop

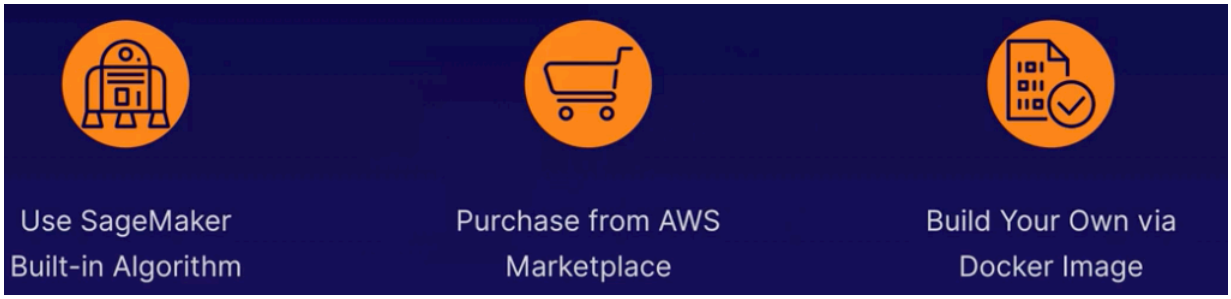
We might skew the model to fulfill our assumptions in the fist place



3 types of Learning:

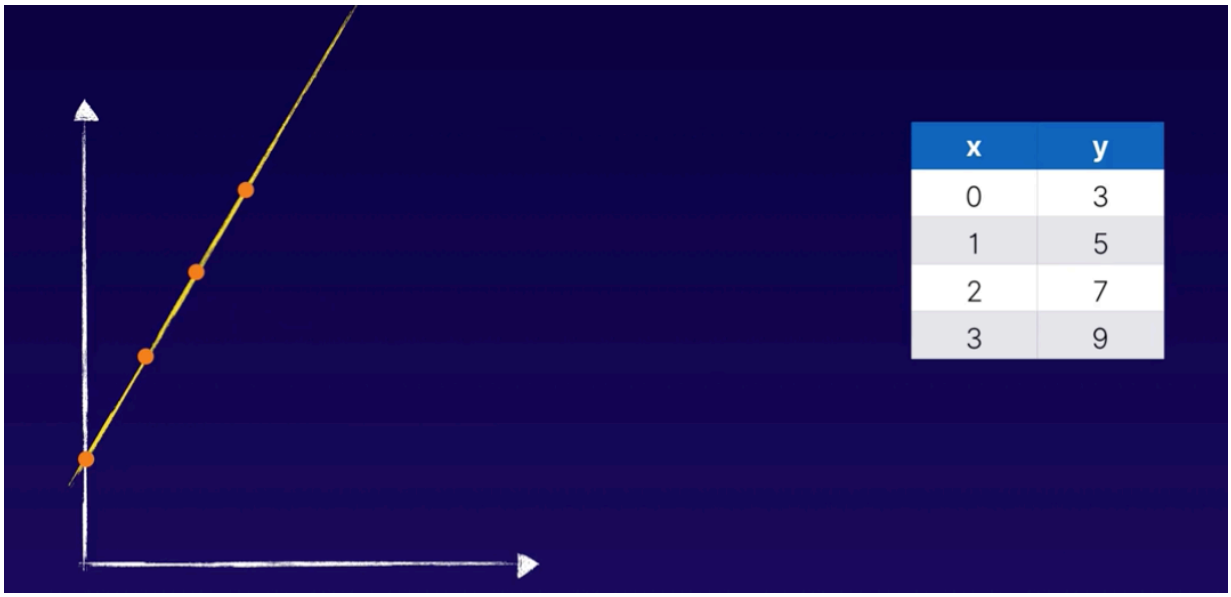
	Supervised Learning	Unsupervised Learning	Reinforcement Learning
Training	Training Data and Testing Data	No Training	How to Maximize Reward
Discrete	Classification	Clustering	Simulation-based Optimization
Continuous	Regression	Reduction of Dimentionality	Autonomous Devices

Algorithms:



Regression

Example: linear equation: y = 2x + 3
Given x, we can figure out y and vice-versa
We can plot a line too.
Seeing values on that line, we can estimate/predict what is the y value further down on that line

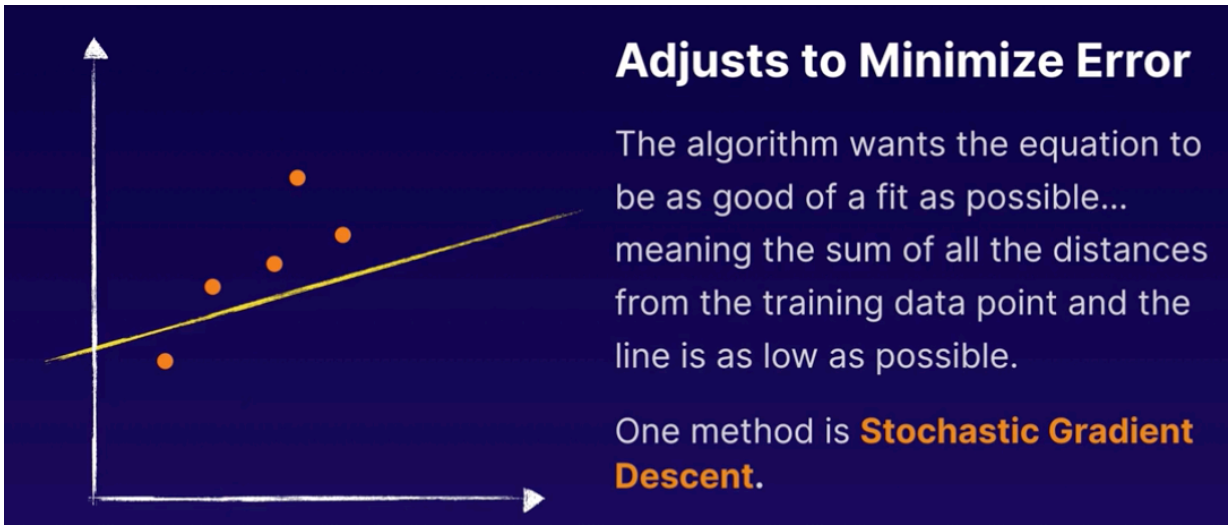


x	y
0	3
1	5
2	7
3	9

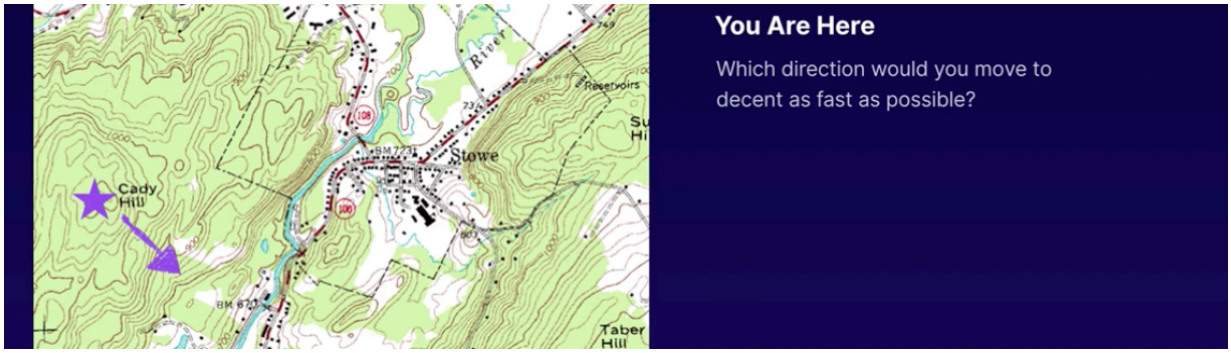
Linear Learner Algorithm

Linear models are supervised learning algorithms for regression, binary classification or multiclass classification problems. You give the model labels (x,y) with x being high-dimensional vector and y is a numeric label. The algorithm learns a linear function, or, for classification problems, a linear threshold function, and maps a vector x to an approximation of label y.

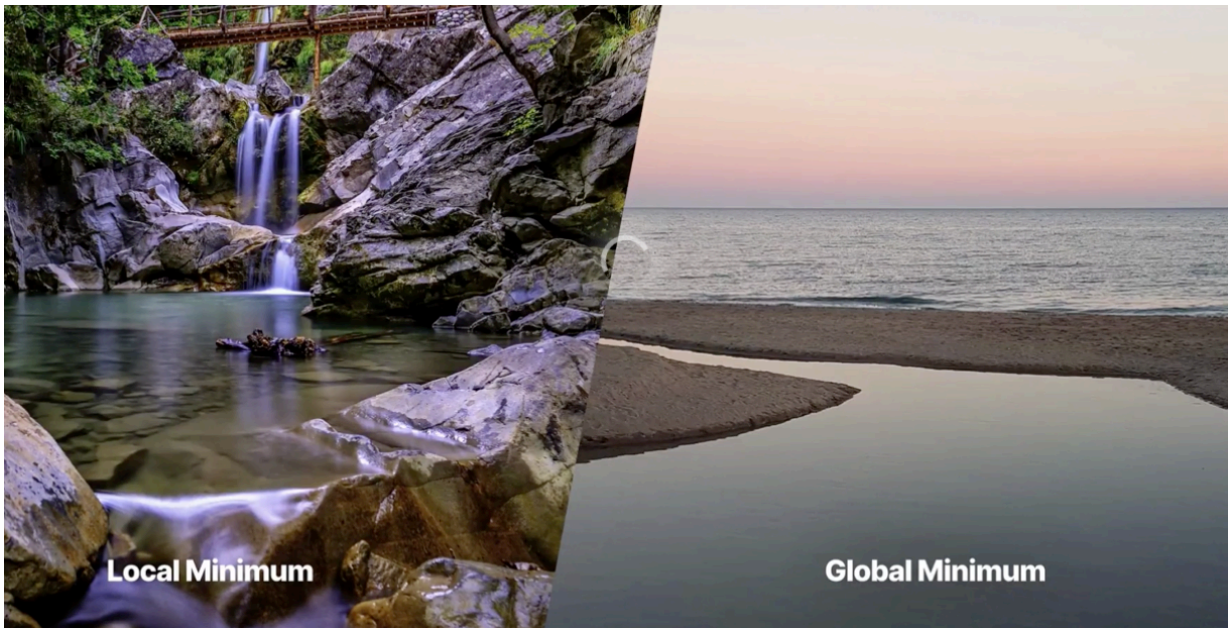
To use this algorithm you need a number or list of numbers which yields some other number...the answer you're after. You can use it to predict a specific value or a threshold for grouping purposes.



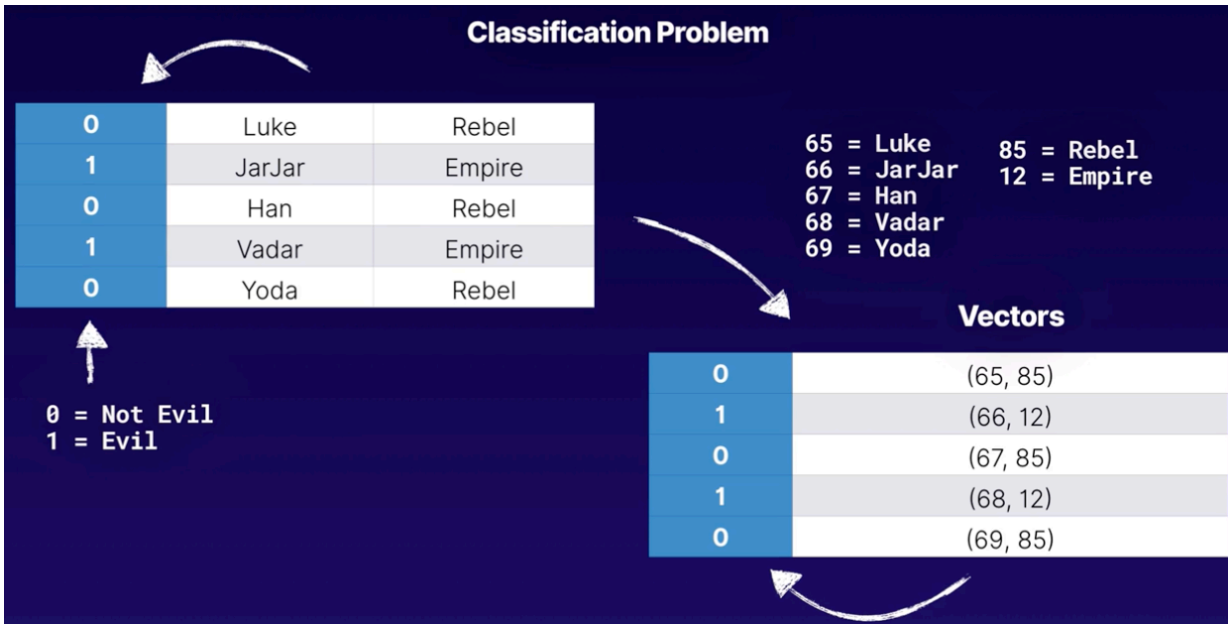
Ex of topographic map to explain Stochastic Gradient Descent
Map is the error.



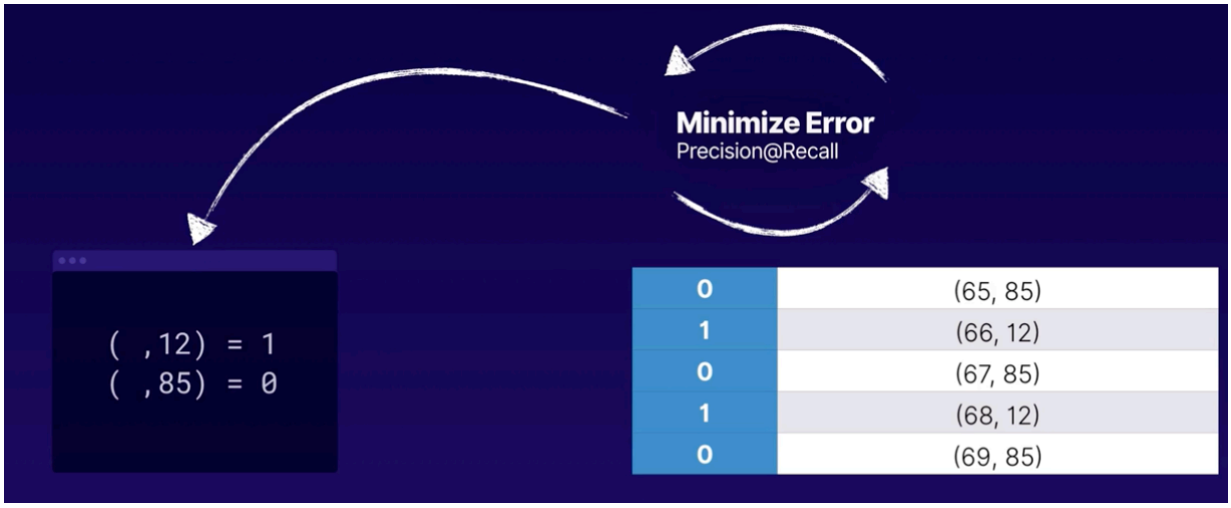
Another idea - idea coming down
Some may be stuck in pools, some may make their way all the way to the ocean

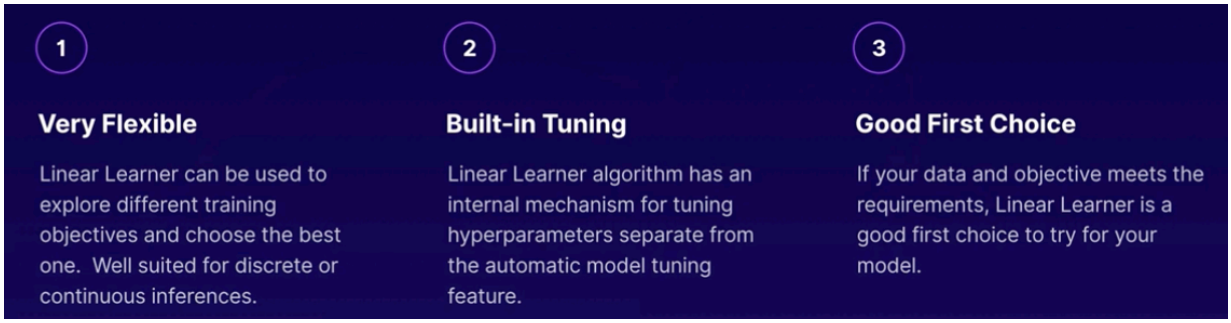


For linear regression, we need numbers -> convert the data and end up with vectors

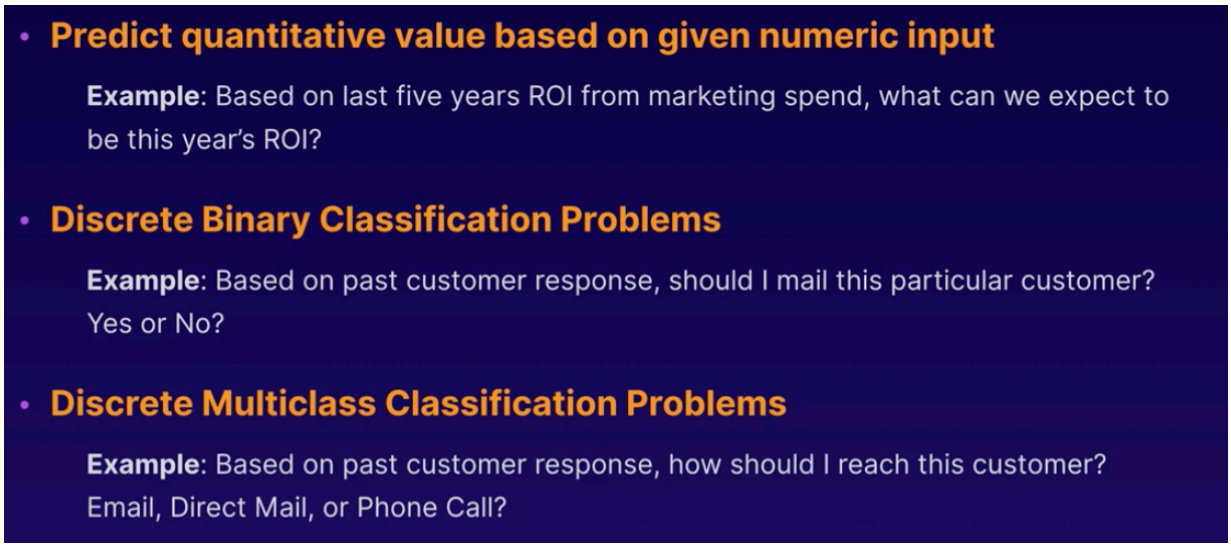


Linear learning will learn through trial/error process, is that the 2nd value in the vector is the most accurate to predict outcome, whether somebody is a 0 or 1

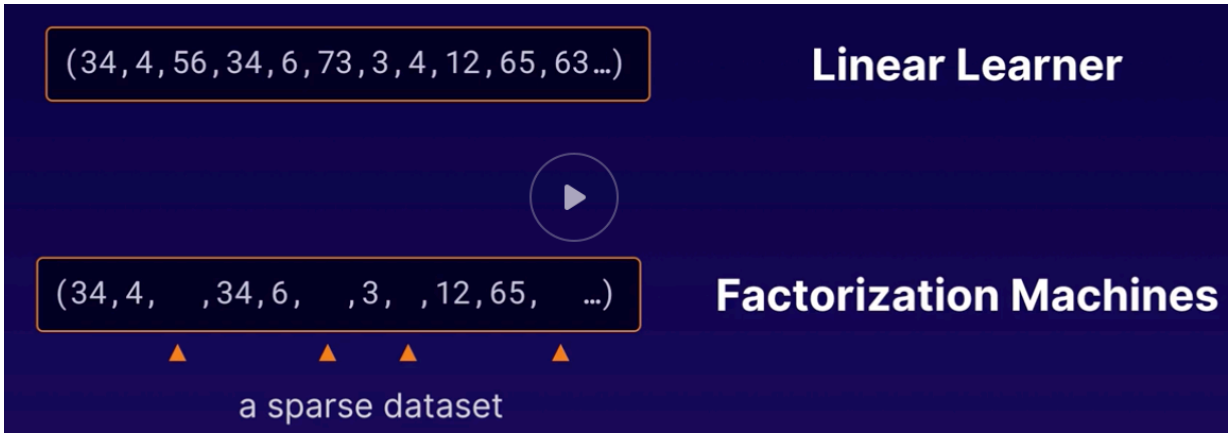


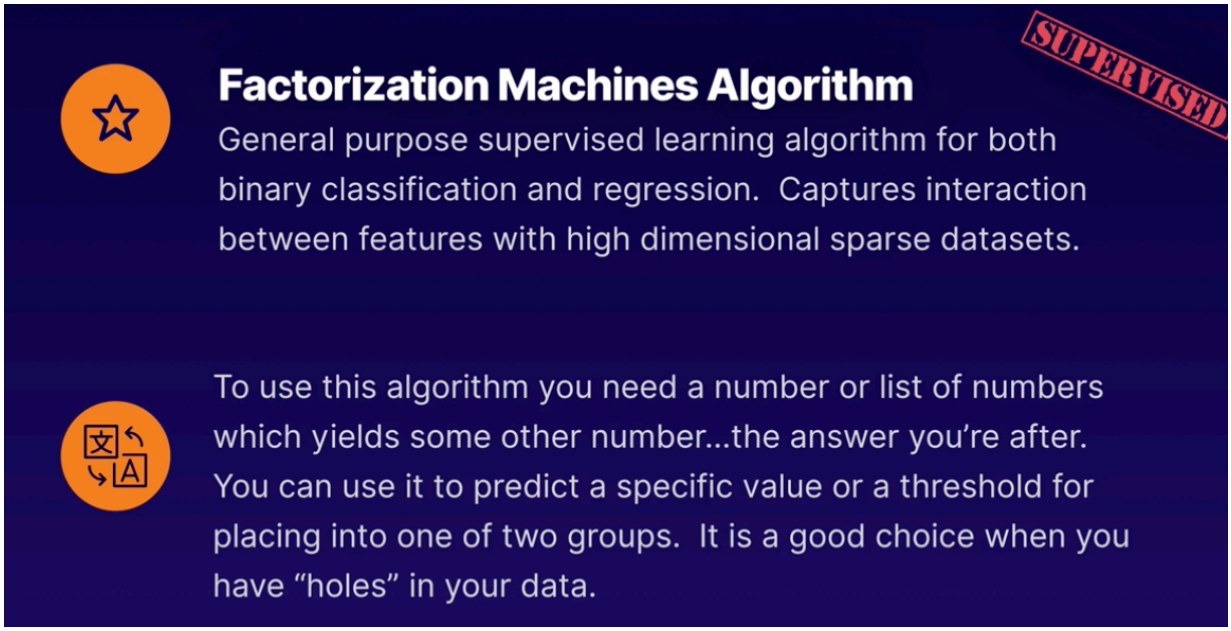


Examples when to use Linear Learner Algorithm - Use Cases:



It works well when we have good continuous data.
Wit sparse dataset, need to adjust this with Factorization Machine

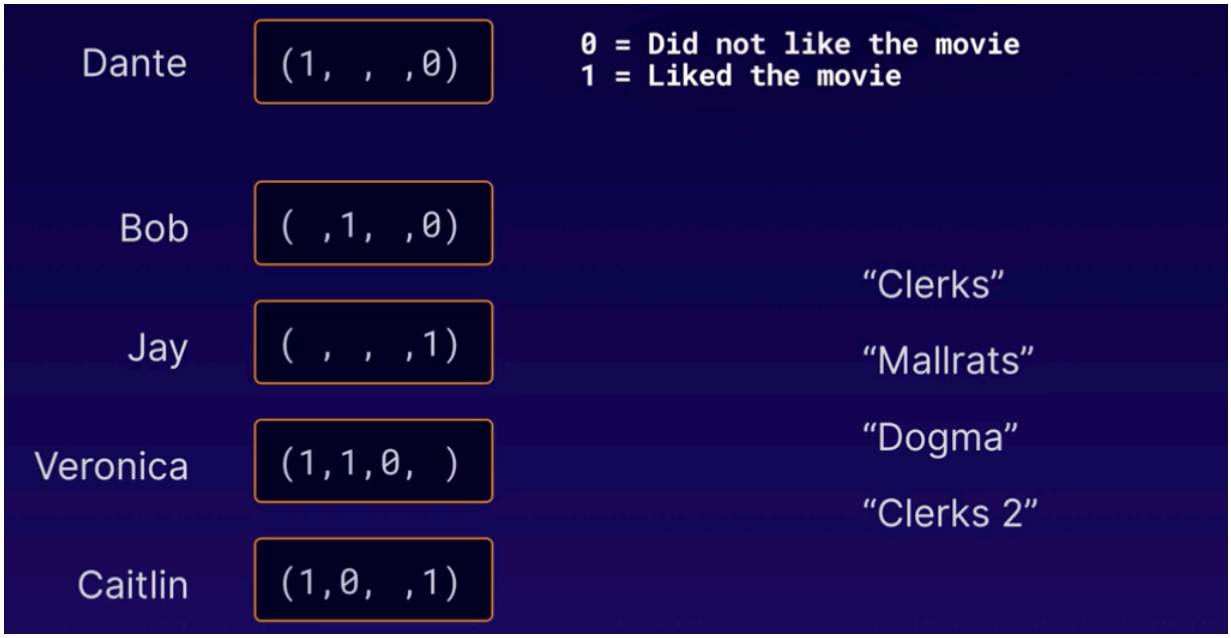




Things to know about Factorization Machines Algorithm

Example: Movie recommendation engine
4 movies watched by 4 people.
Nobody has watched all movies —> perfect situation for a recommender engine

Objective: We want to find a movie that Dante is going to like

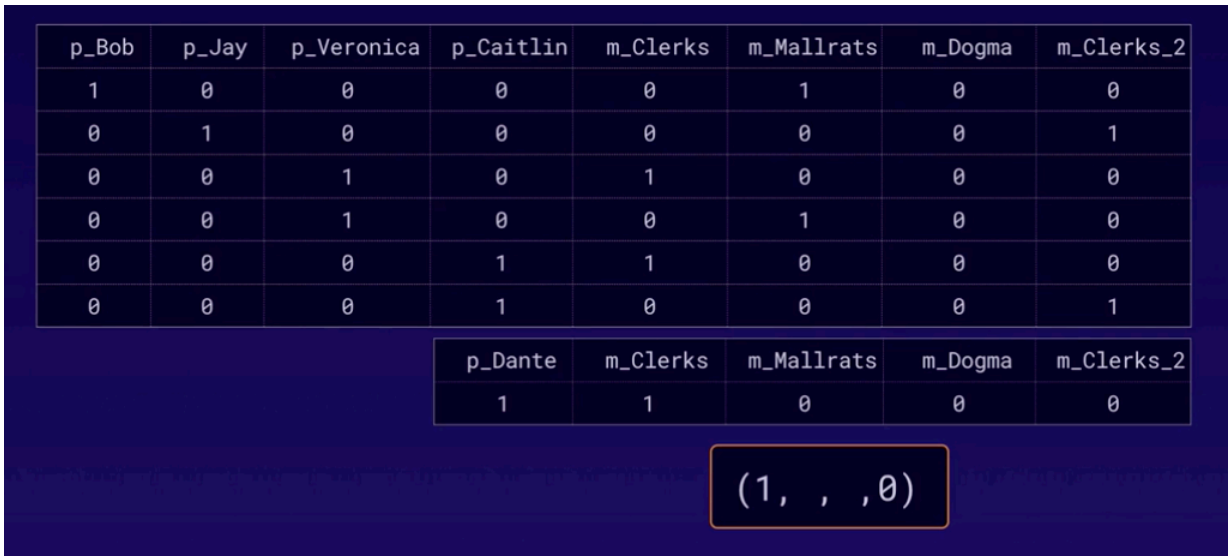


=> do one hot encoding that creates a matrix for us to evaluate with movies and people that like them

p_Bob	p_Jay	p_Veronica	p_Caitlin	m_Clerks	m_Mallrats	m_Dogma	m_Clerks_2
1	0	0	0	0	1	0	0
0	1	0	0	0	0	0	1
0	0	1	0	1	0	0	0
0	0	1	0	0	1	0	0
0	0	0	1	1	0	0	0
0	0	0	1	0	0	0	1

```
1,0,0,0,0,1,0,0
0,1,0,0,0,0,0,1
0,0,1,0,1,0,0,0
0,0,1,0,0,1,0,0
0,0,0,1,1,0,0,0
0,0,0,1,0,0,0,1
```

To note if we had 100,000 viewers and 10,000 films, we would end up with a really huge matrix, filled for the most with 0
That is a huge waste and why Factorization Machine does not support CSV => rather they use recordIO protobuff with Float32 Tensor



Start building a profile of Dante's movies preferences vs other's preference, and summing that

p_Bob	p_Jay	p_Veronica	p_Caitlin	m_Clerks	m_Mallrats	m_Dogma	m_Clerks_2
1	0	0	0	0	1	0	0
0	1	0	0	0	0	0	1
0	0	1	0	1	0	0	0
0	0	1	0	0	1	0	0
0	0	0	1	1	0	0	0
0	0	0	1	0	0	0	1

p_Dante	m_Clerks	m_Mallrats	m_Dogma	m_Clerks_2
1	1	0	0	0

Difference between Dante and Other People

m_Clerks	m_Mallrats	m_Dogma	m_Clerks_2
-1	1	0	0
-1	0	0	1
0	0	0	0
-1	1	0	0
0	0	0	0
-1	0	0	1
Sum	-4	2	0	2

We want anything positive

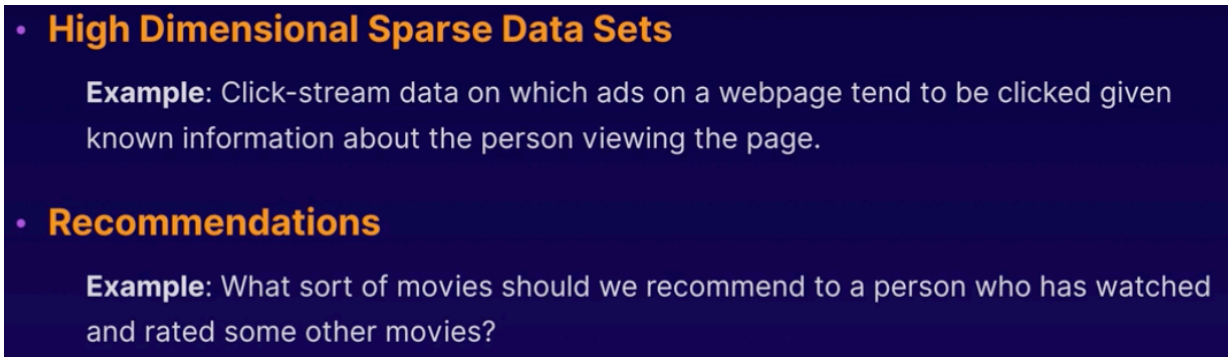
Difference between Dante and Other People

m_Clerks	m_Mallrats	m_Dogma	m_Clerks_2
-1	1	0	0
-1	0	0	1
0	0	0	0
-1	1	0	0
0	0	0	0
-1	0	0	1
Sum	-4	2	0	Nope

Dante (1, , ,0)

I recommend “Mallrats”.

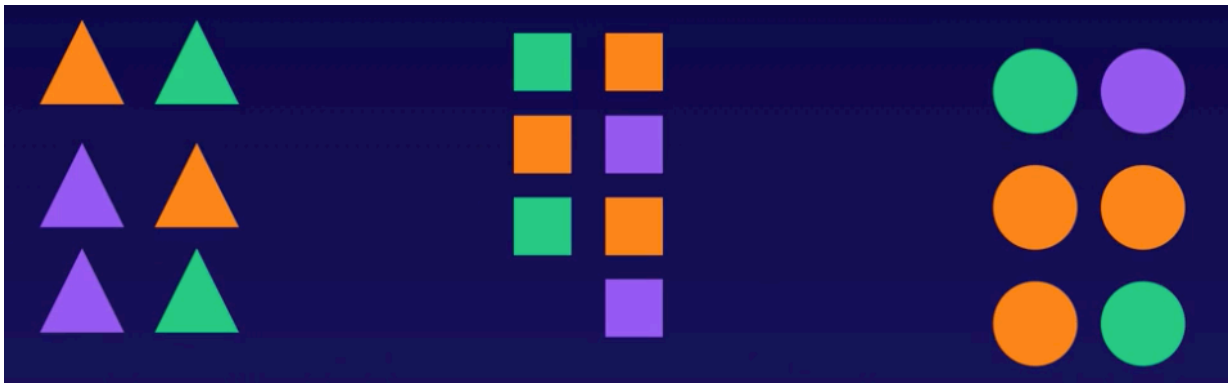
Use case for Factorization Machine Algorithm:



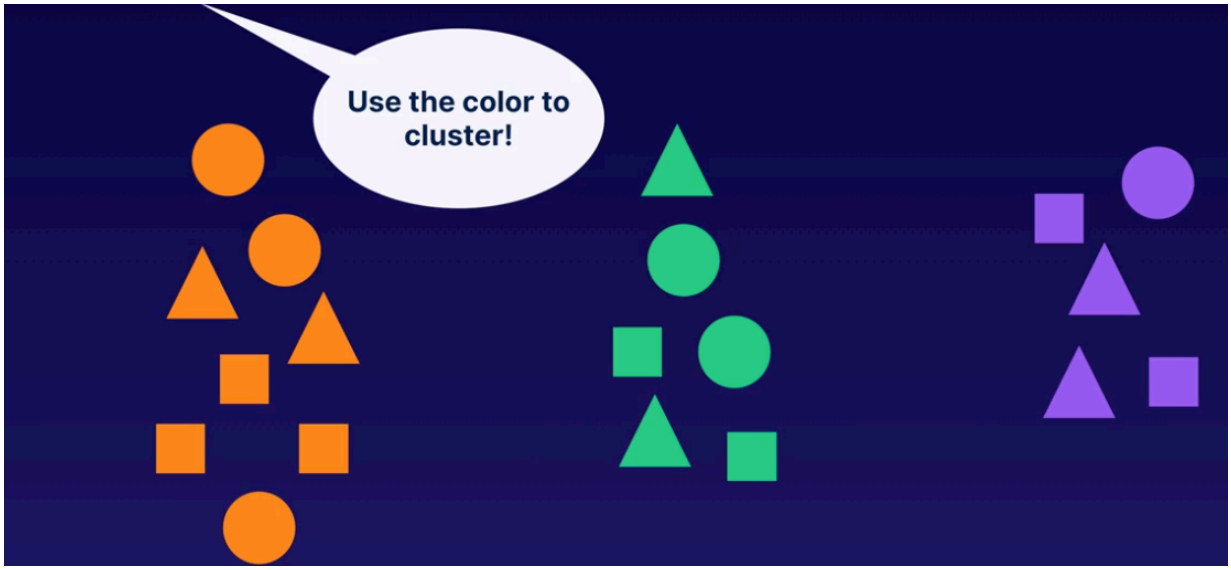
Clustering

Unsupervised - no need for training data

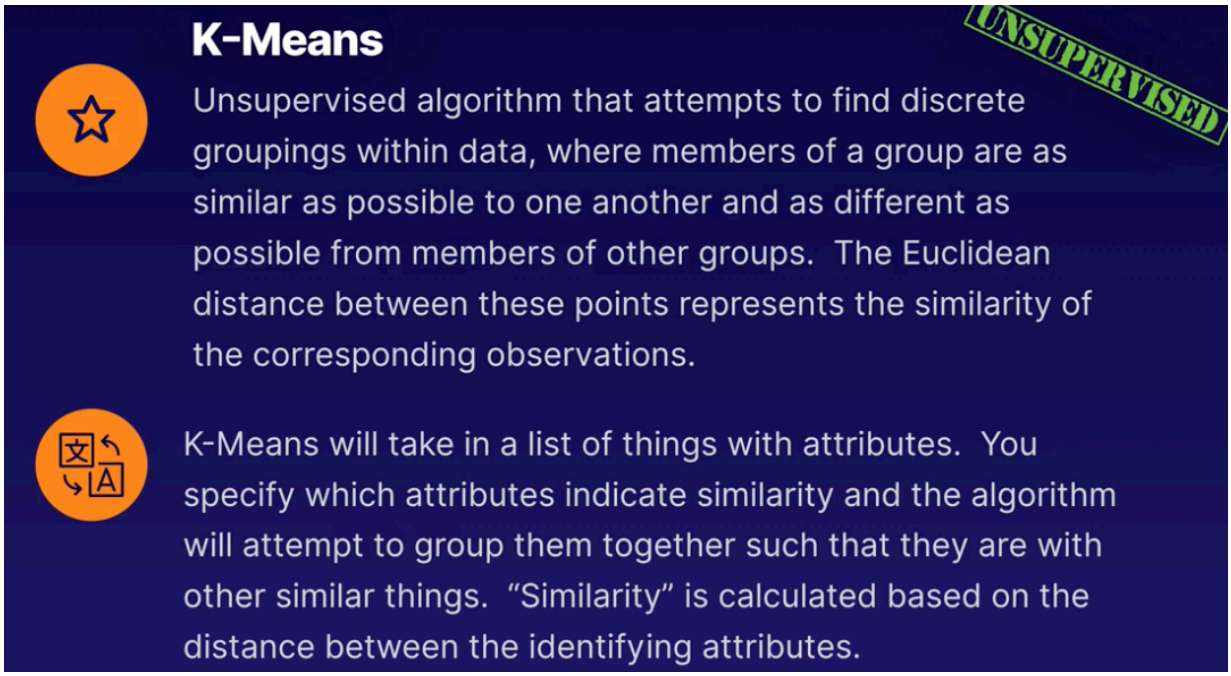
Specify which column a clustering algorithm need to pay attention to to group the data together



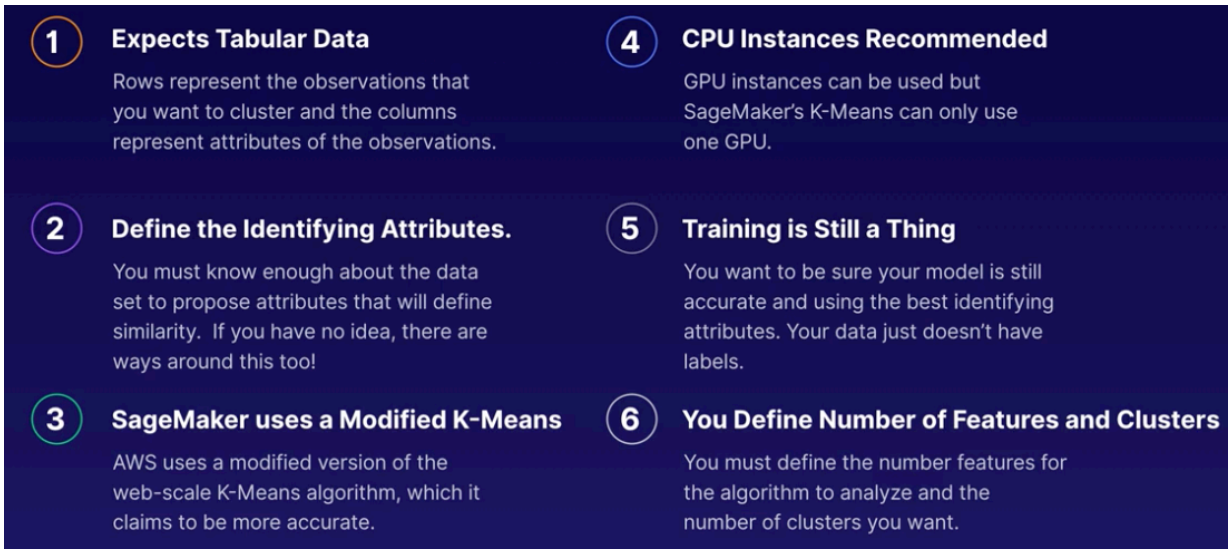
We could also cluster based on color



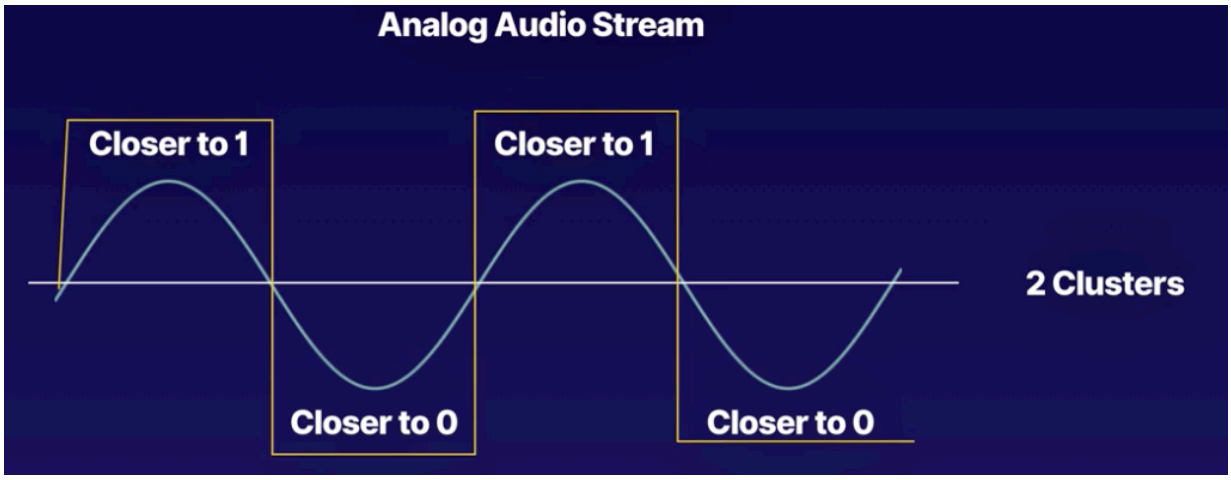
Popular clustering algorithm: K-Means



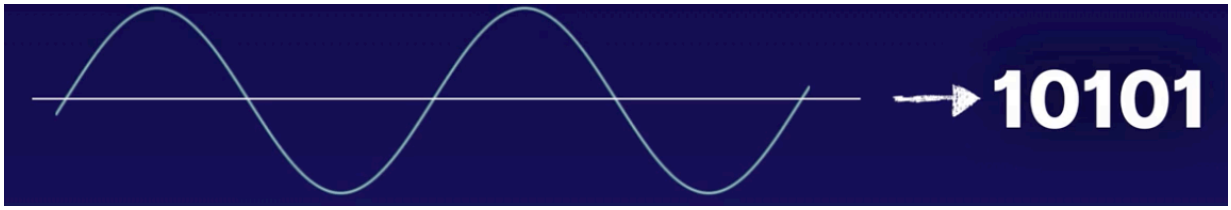
SageMaker K-means - things to know:



Example of application: audio signal processing

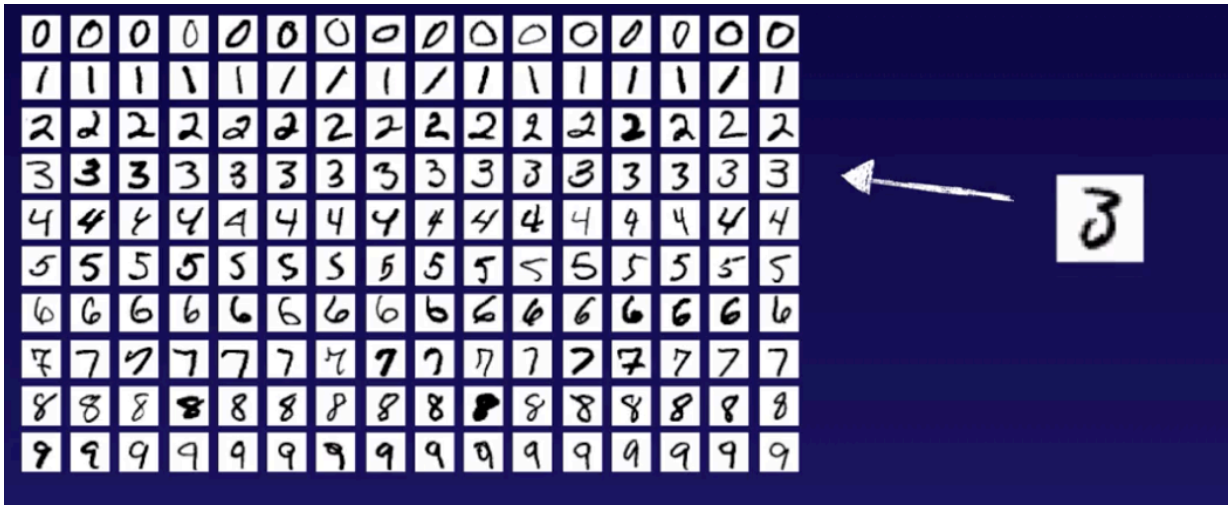


Translate audio signal into digital signal with 0 and 1



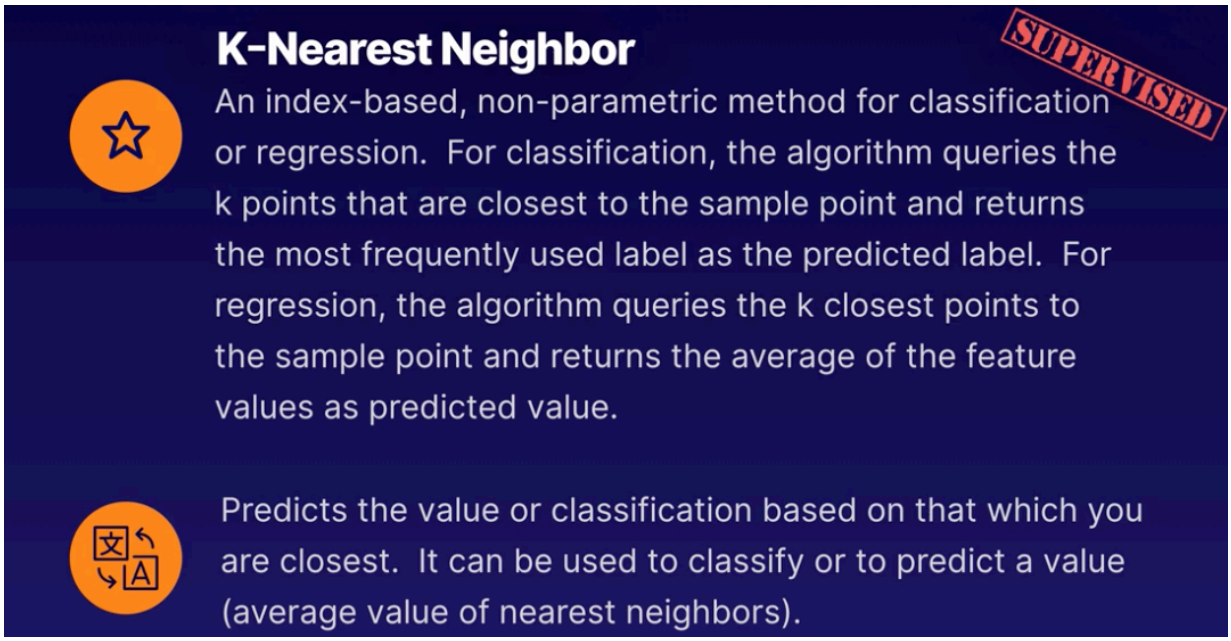
The digitization process clips a bit of the fidelity from the analog

Other example: cluster the MNIST dataset
Store coordinates of white/black pixels and build vectors

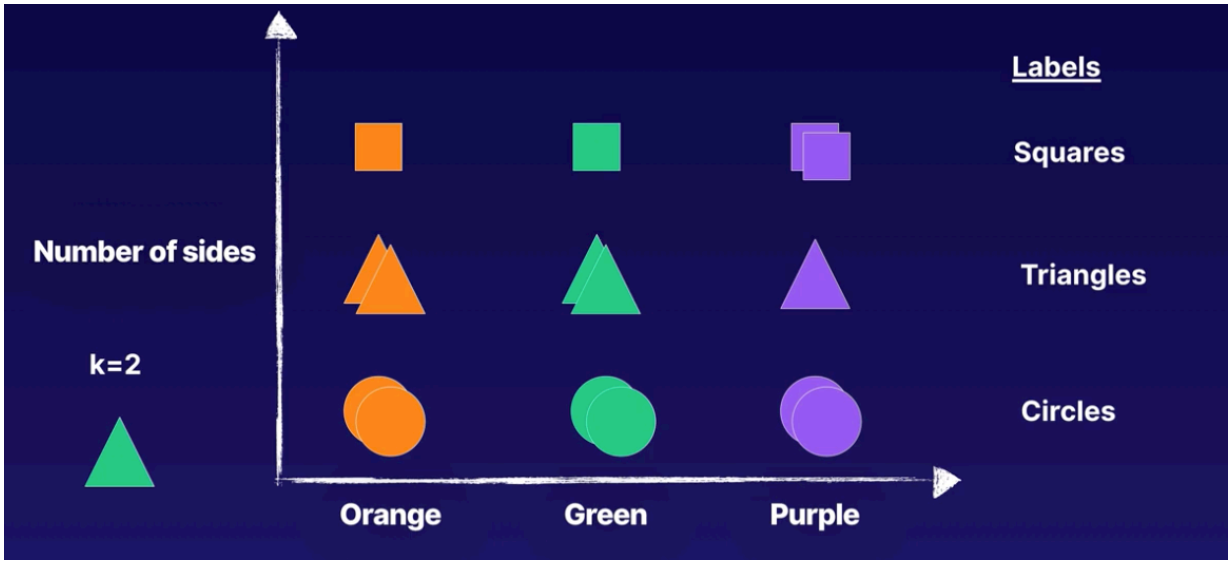


Classification

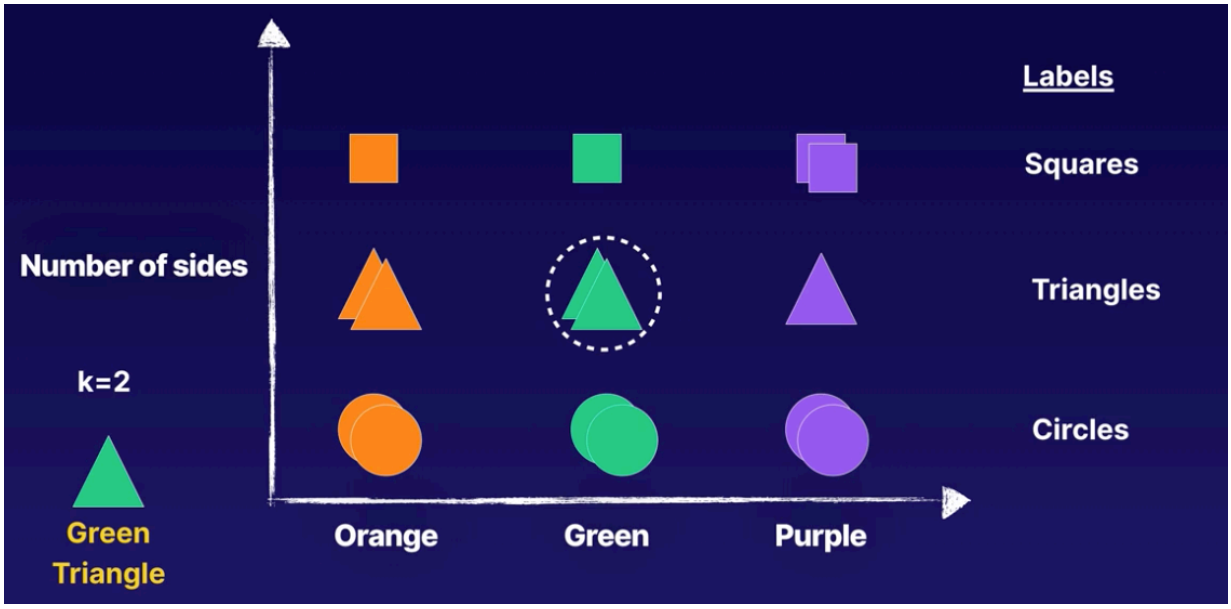
Popular classification algorithm in SM: K-Nearest Neighbor

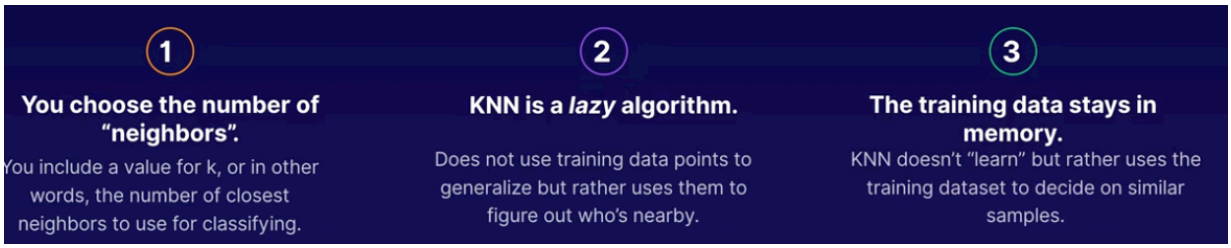


Example:

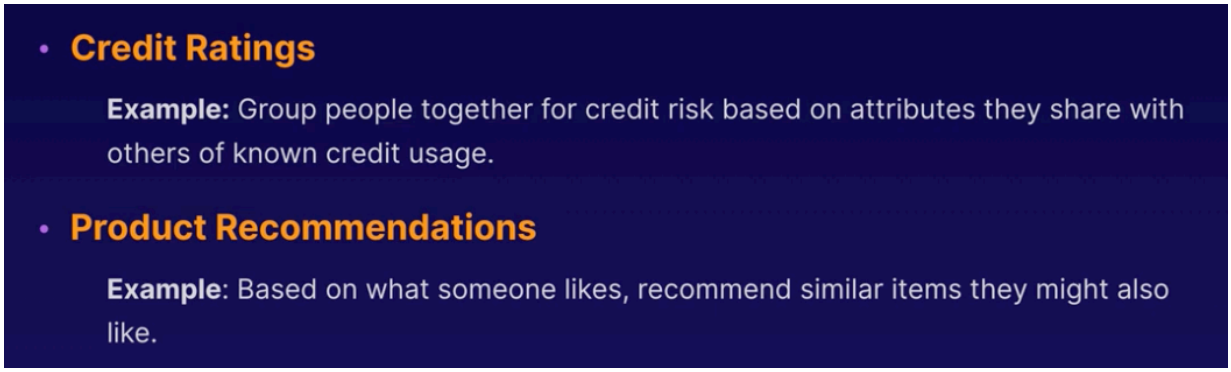


Hyper parameter: k = 2
Find the 2 nearest neighbors to this input shape (a triangle)





Use cases for K-nearest neighbors:



K-nearest neighbor is a bit a matter of stereotyping
So be aware of Bias

Example when classification has gone wrong:

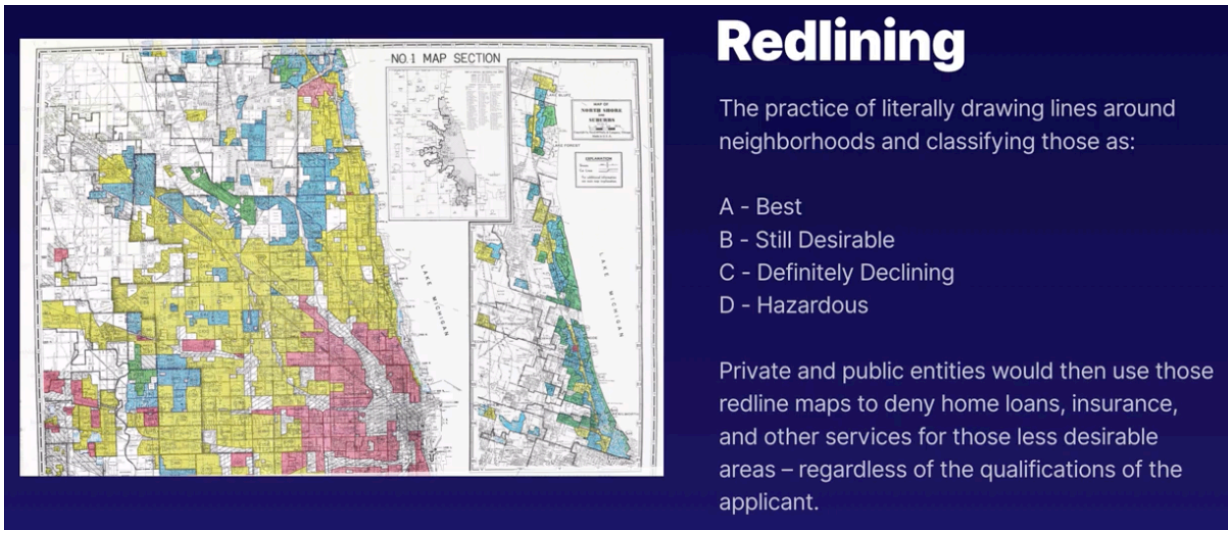


Image Analysis