Features of Success

Finding Structure In High Dimensional Data

CHIDATA

Recap: Simulation v.s. Model Fitting

Fitting: Observe xi, yi, **infer theta**

Regression Problems -> Continuous Y

Classification Problems -> Discrete Y

CHIDATA

Recap: High Dimensional Data

"More Observations Than Individuals"

Stocks: 1000s of securities, Millions of trades

Genomics: 6.4 bn base pairs, 100s Clinical Population

Natural Language: Inf possible word sequences, Millions of Documents

CHIDATA

Problem: Actually Pretty Common!

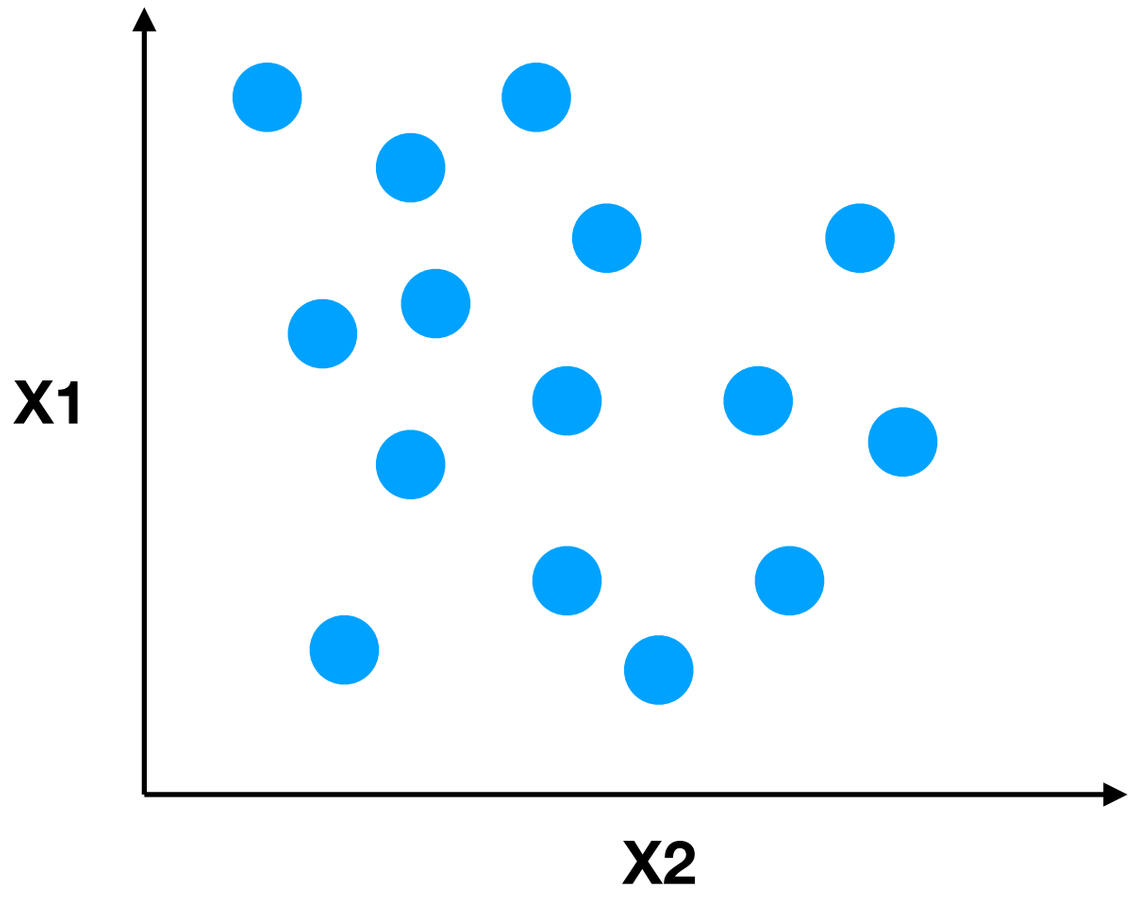
Have to *assume* structure in the data

Low dimensional structure: Large numbers of features are rarely completely independent from each other

CHIDATA

Two Independent Features

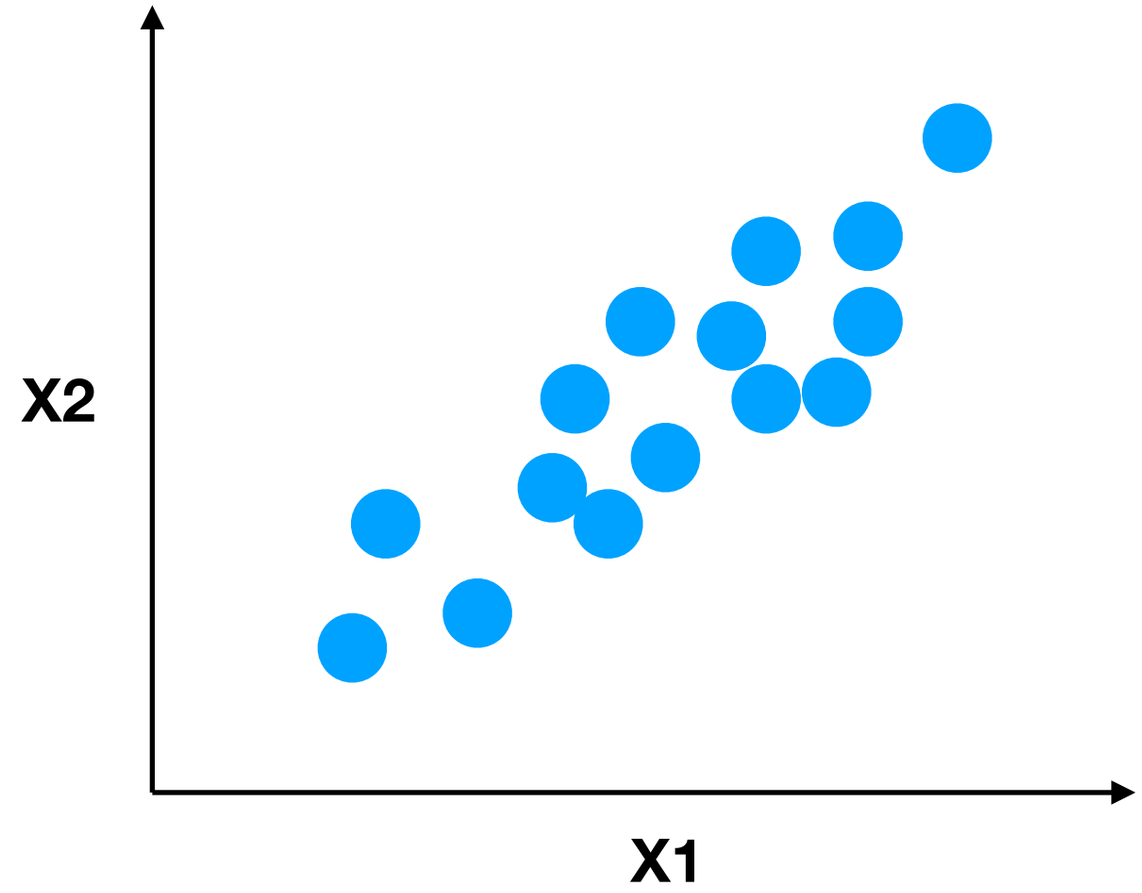
“2 dimensions worth of information”



CHIDATA

Two Dependent Features

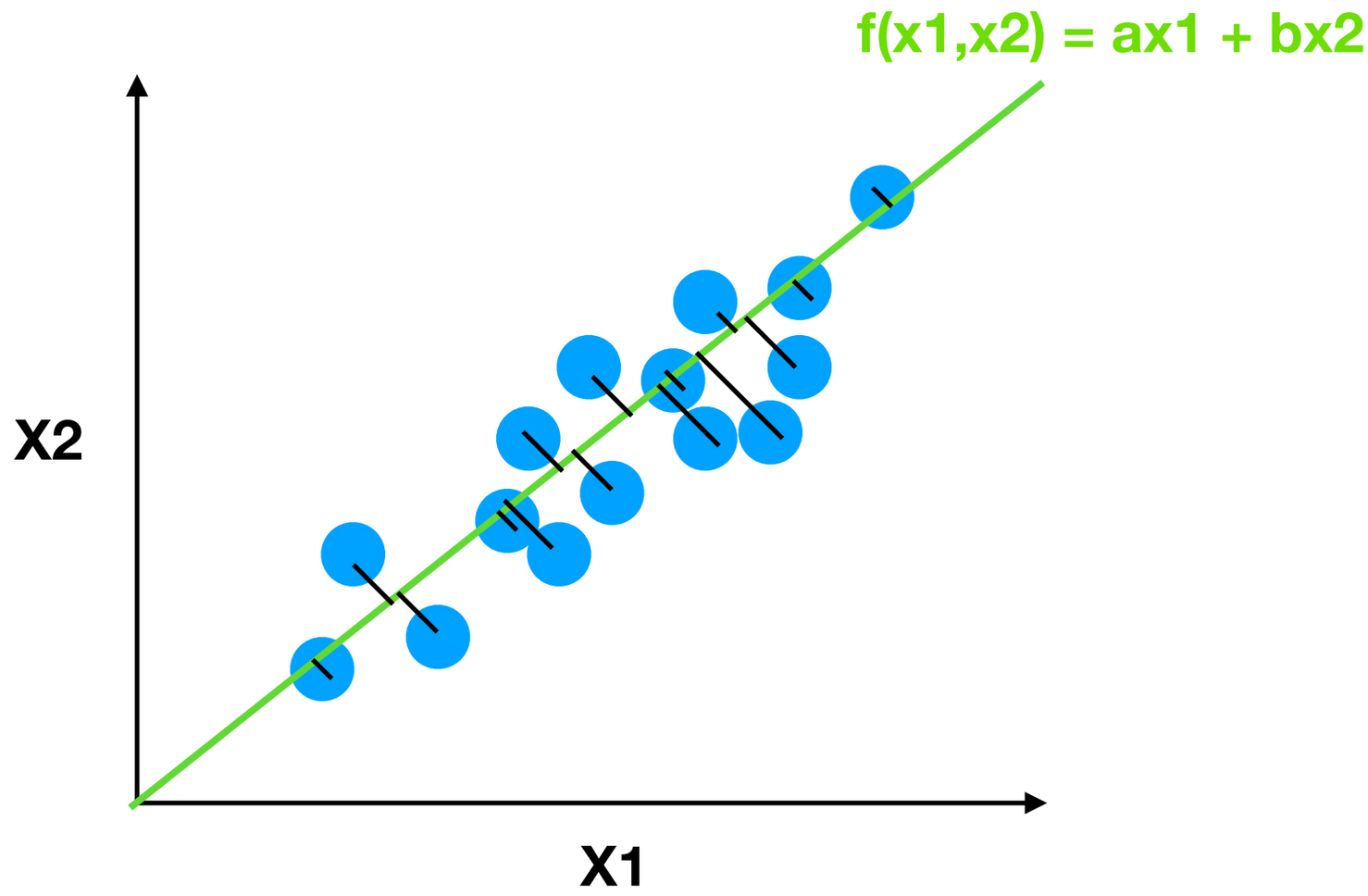
Effectively 1 dimension of information



Intuition: you could use x1 to estimate x2

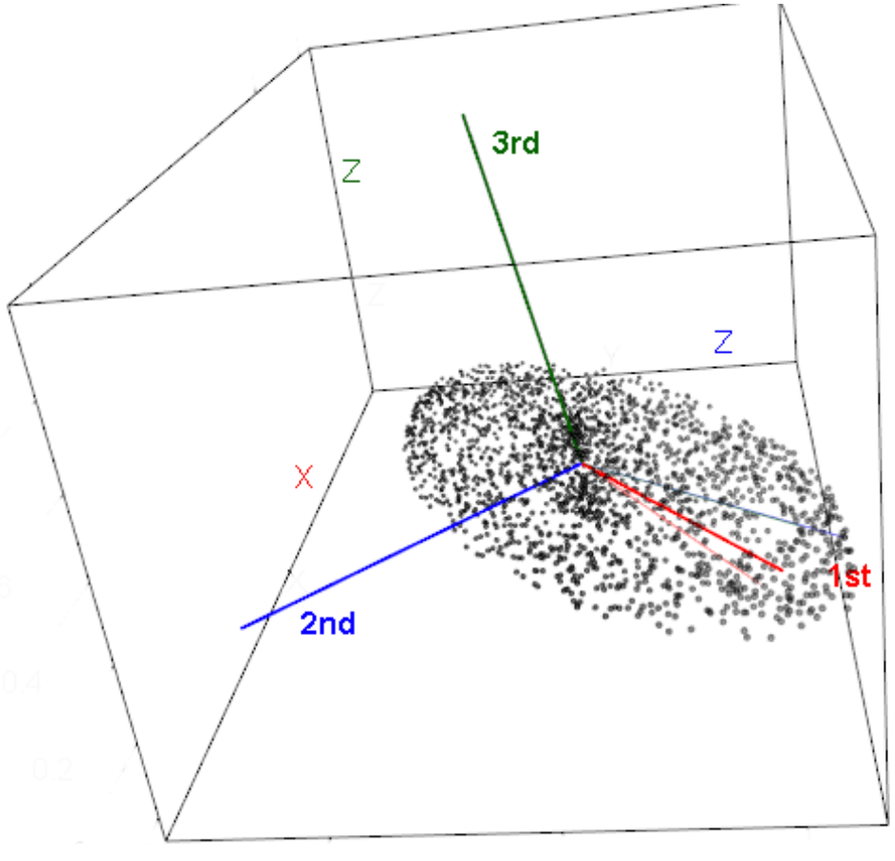
CHIDATA

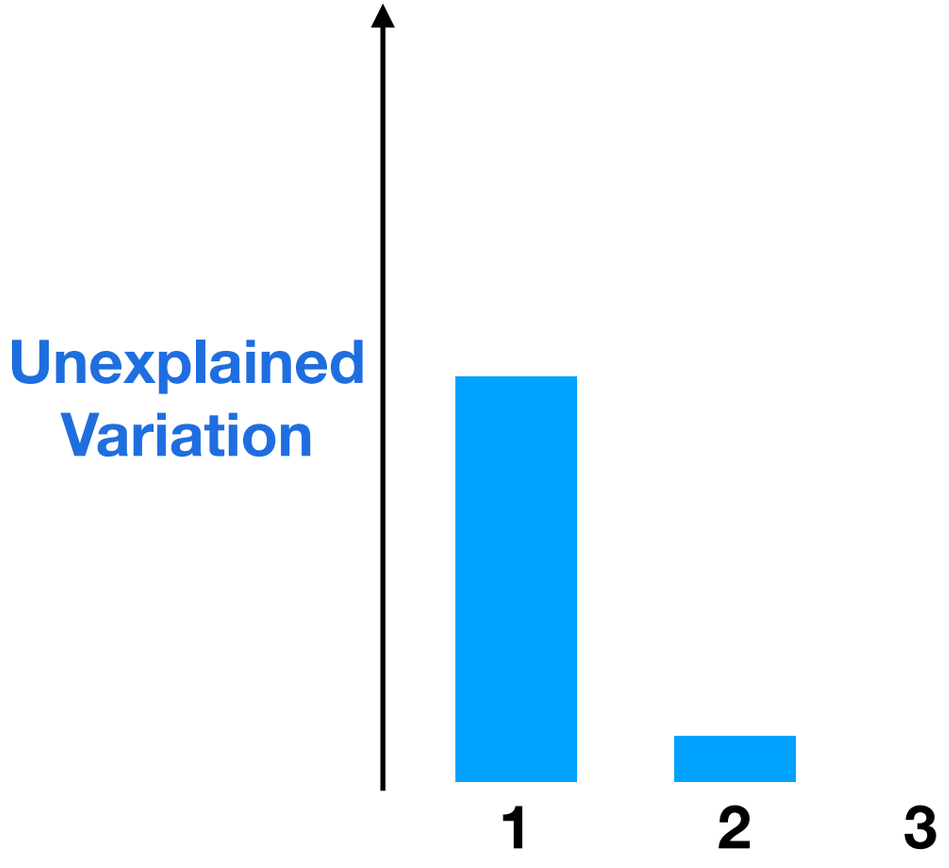
Two Dependent Features



Principal Component Analysis

"PCA": Project features onto a d-dimension plane that explains as much variation as possible.





CHIDATA

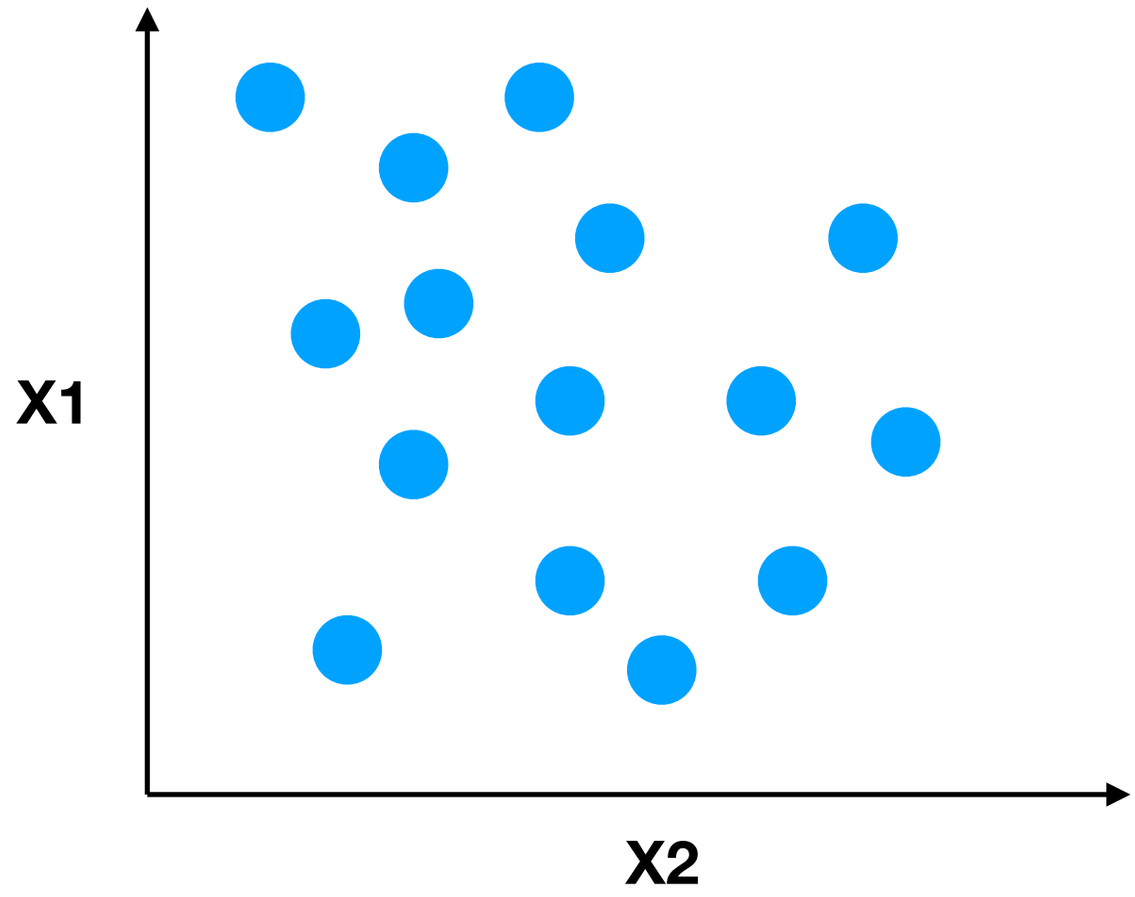
Problem: Actually Pretty Common!

Have to *assume* structure in the data

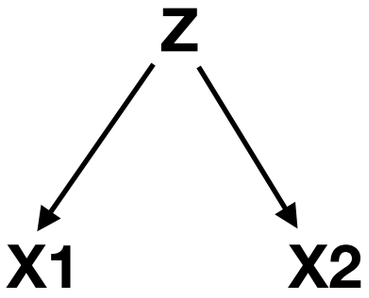
Latent Variable Structure: Observations are governed by an unobserved variable.

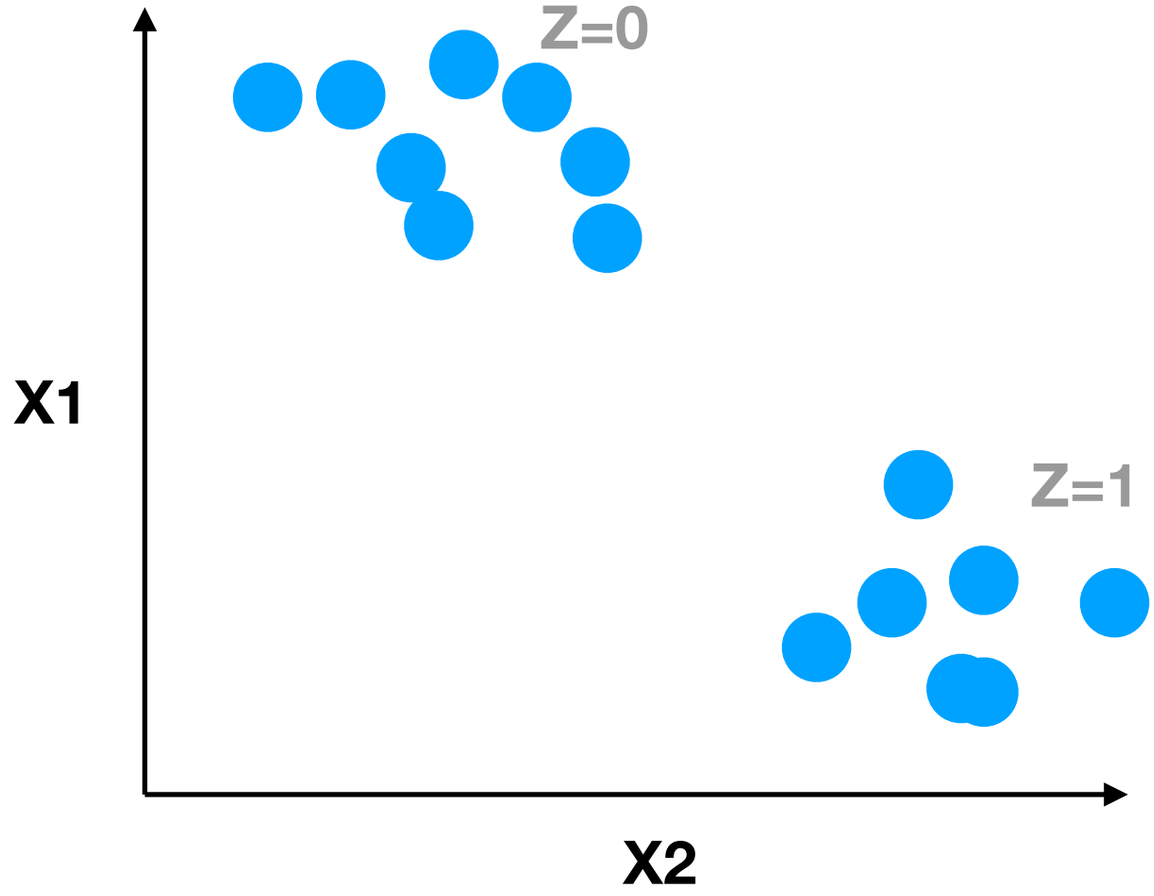
CHIDATA

Two Independent Features



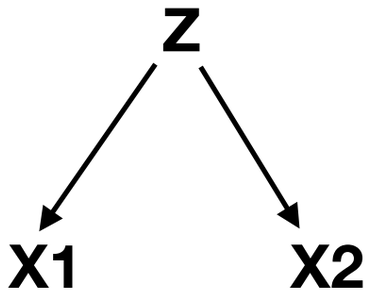
A “Latent” Feature

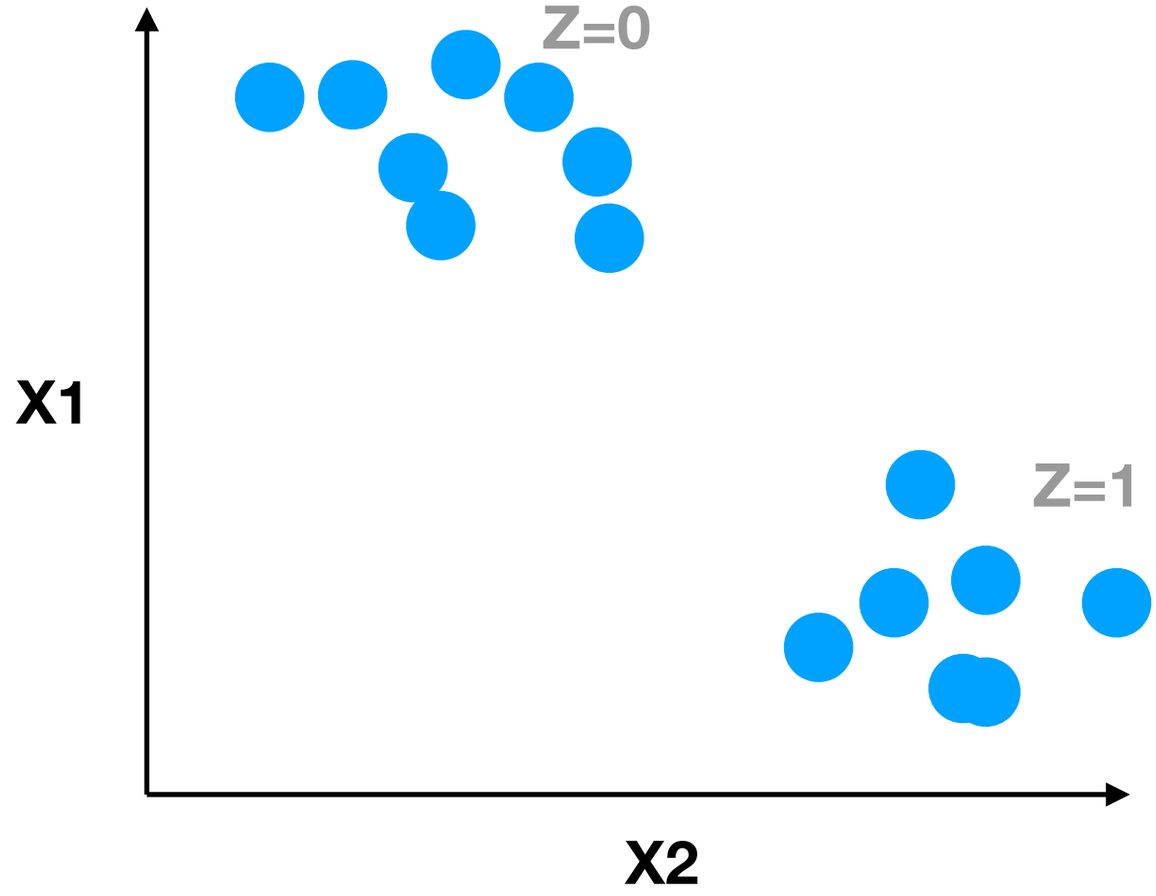




If we knew Z, we could separate out different populations of data

A “Mixture” Model

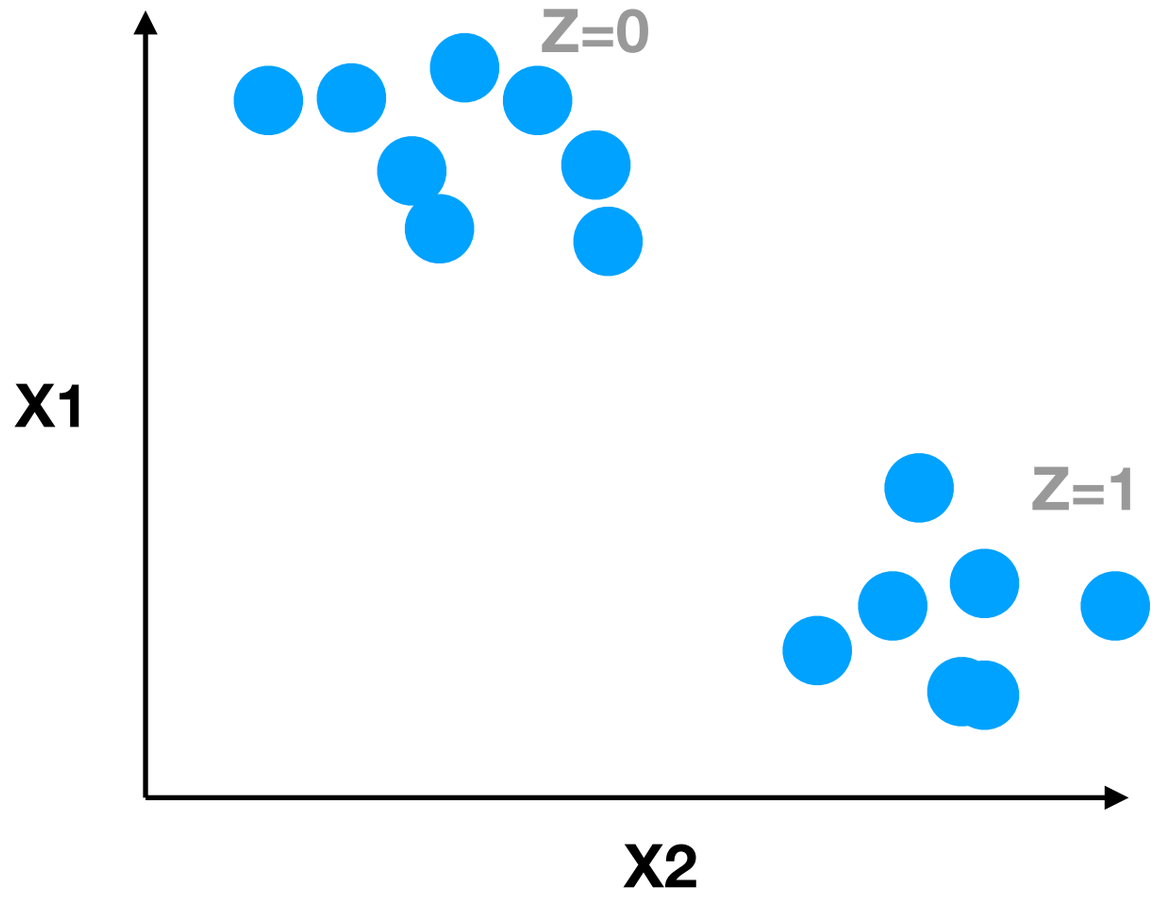




If we knew Z, we could separate out different populations of data

CHIDATA

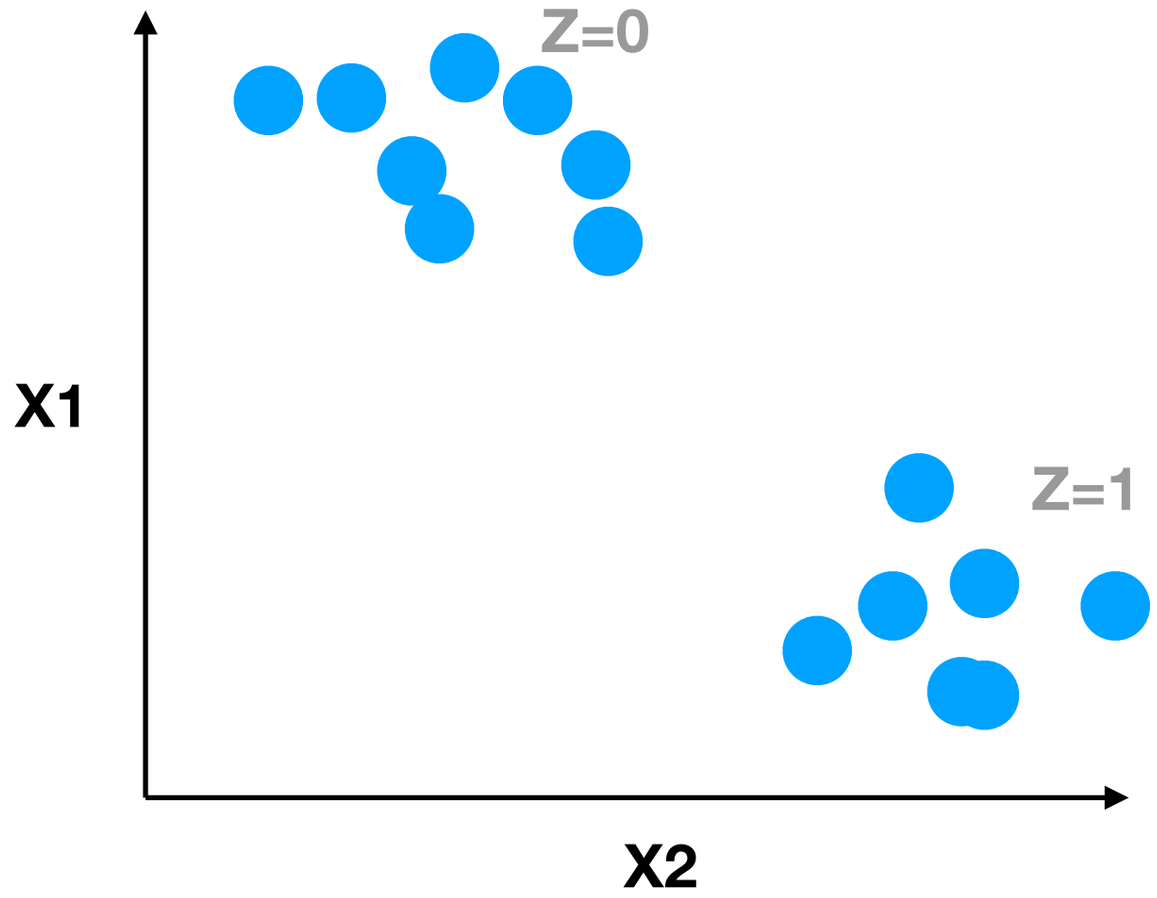
A “Latent” Feature



How do we infer z from the data?

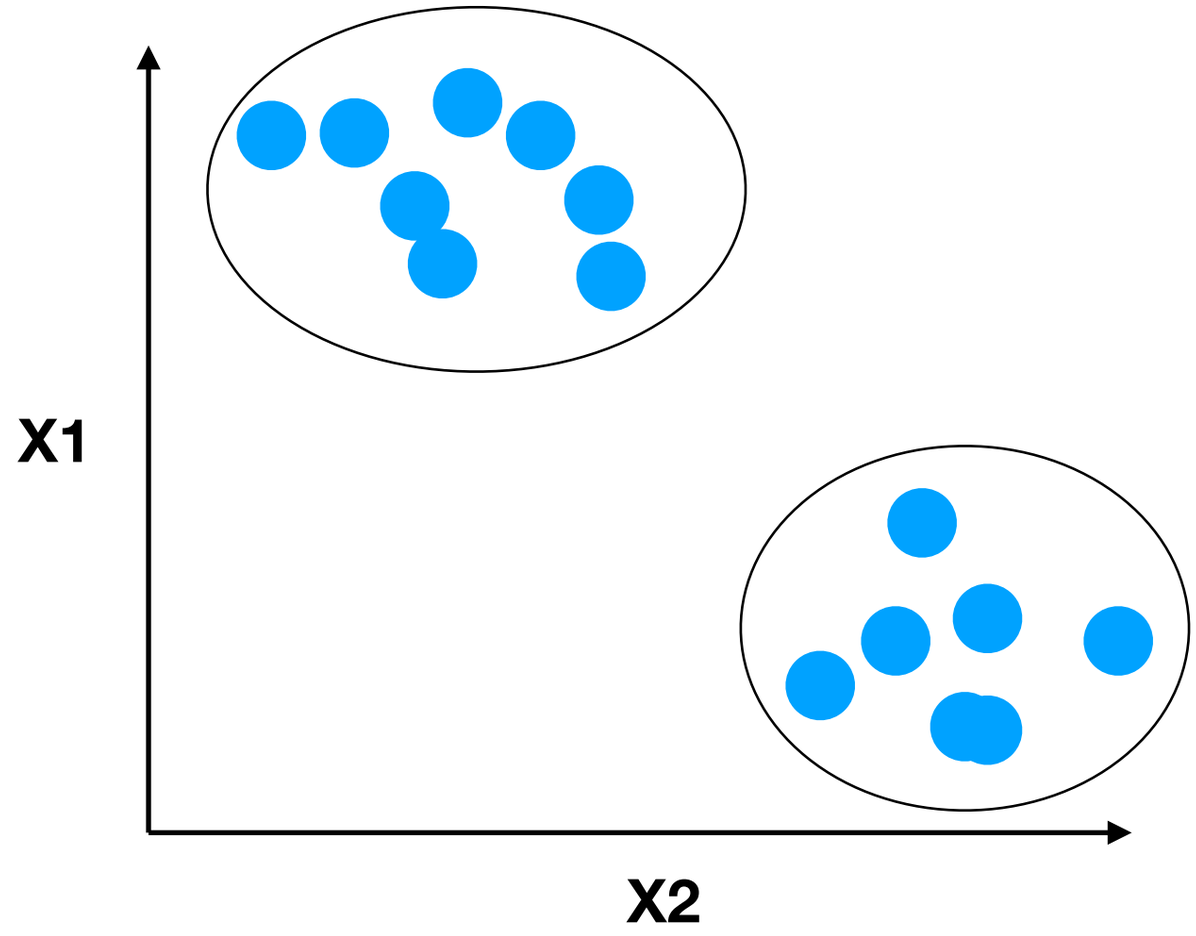
CHIDATA

A “Latent” Feature



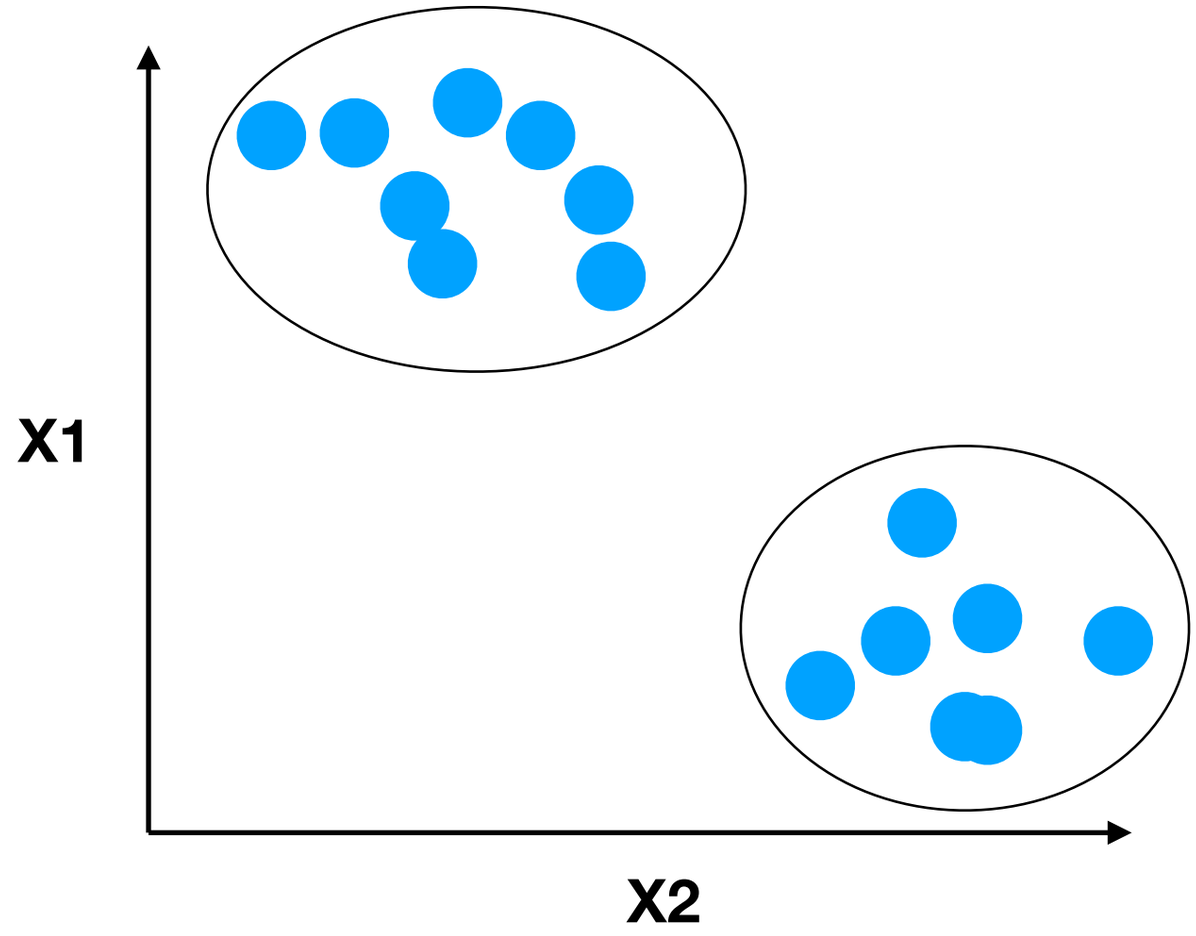
How do we infer z from the data?

Clustering



Algorithms that identify subpopulations in data

Clustering

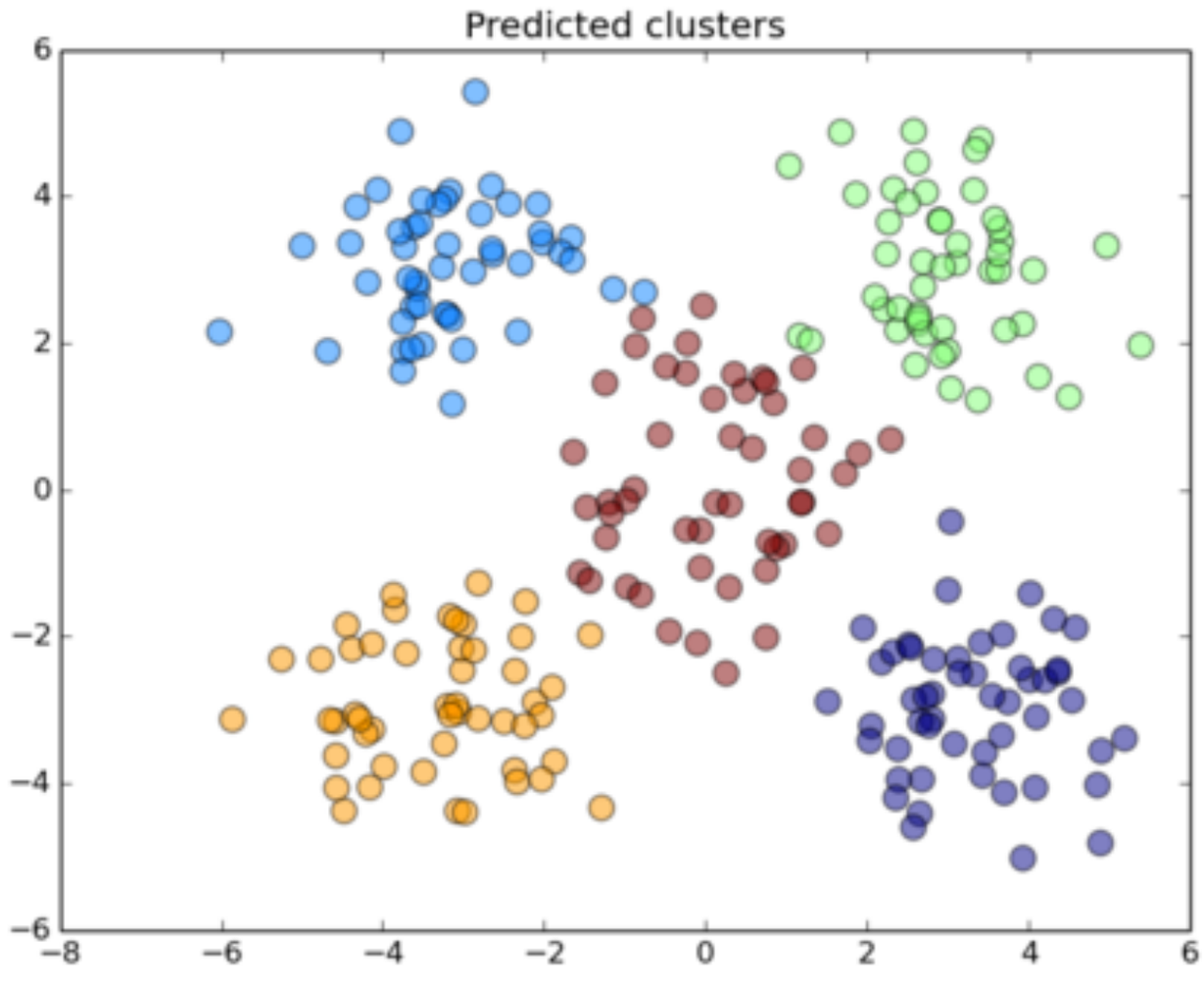


Algorithms that identify subpopulations in data

CHIDATA

K-Means Algorithm

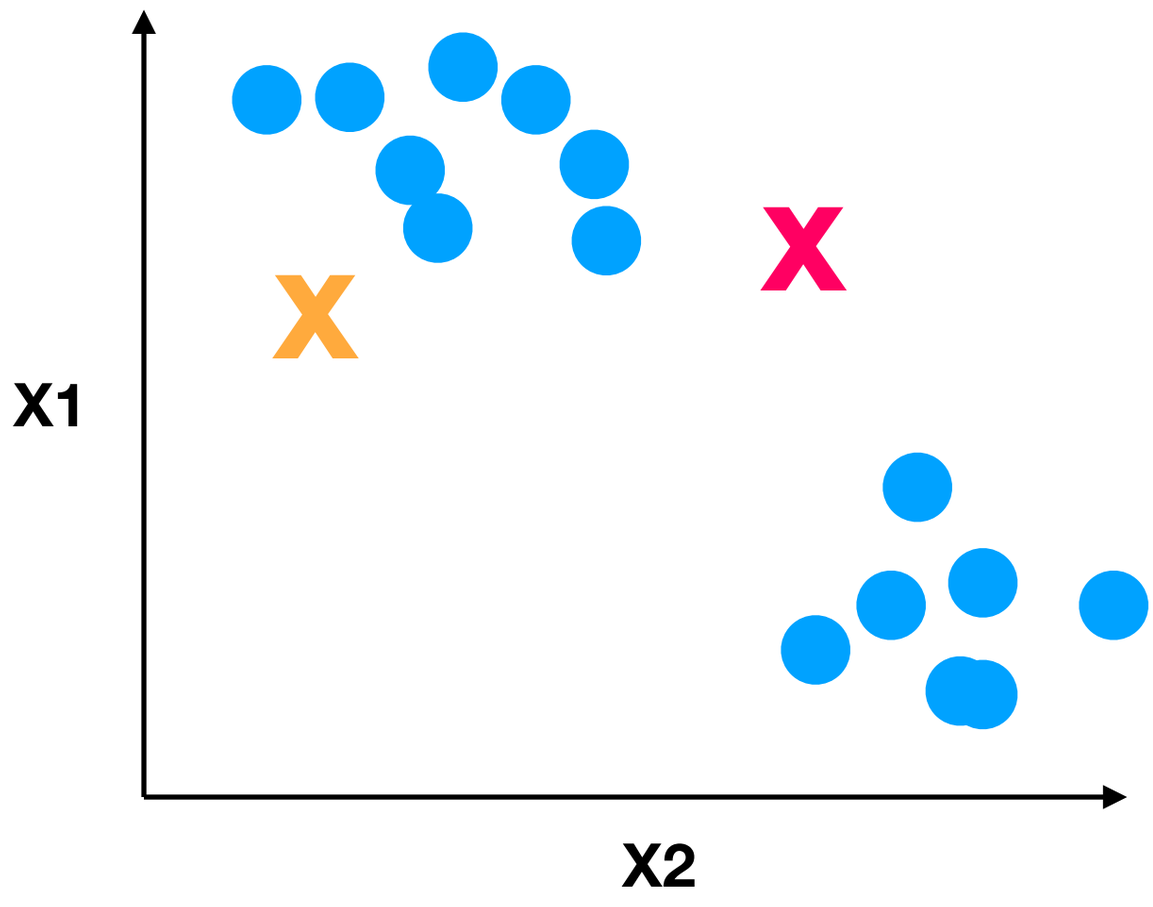
An algorithm for finding clusters in data



CHIDATA

K-Means Algorithm

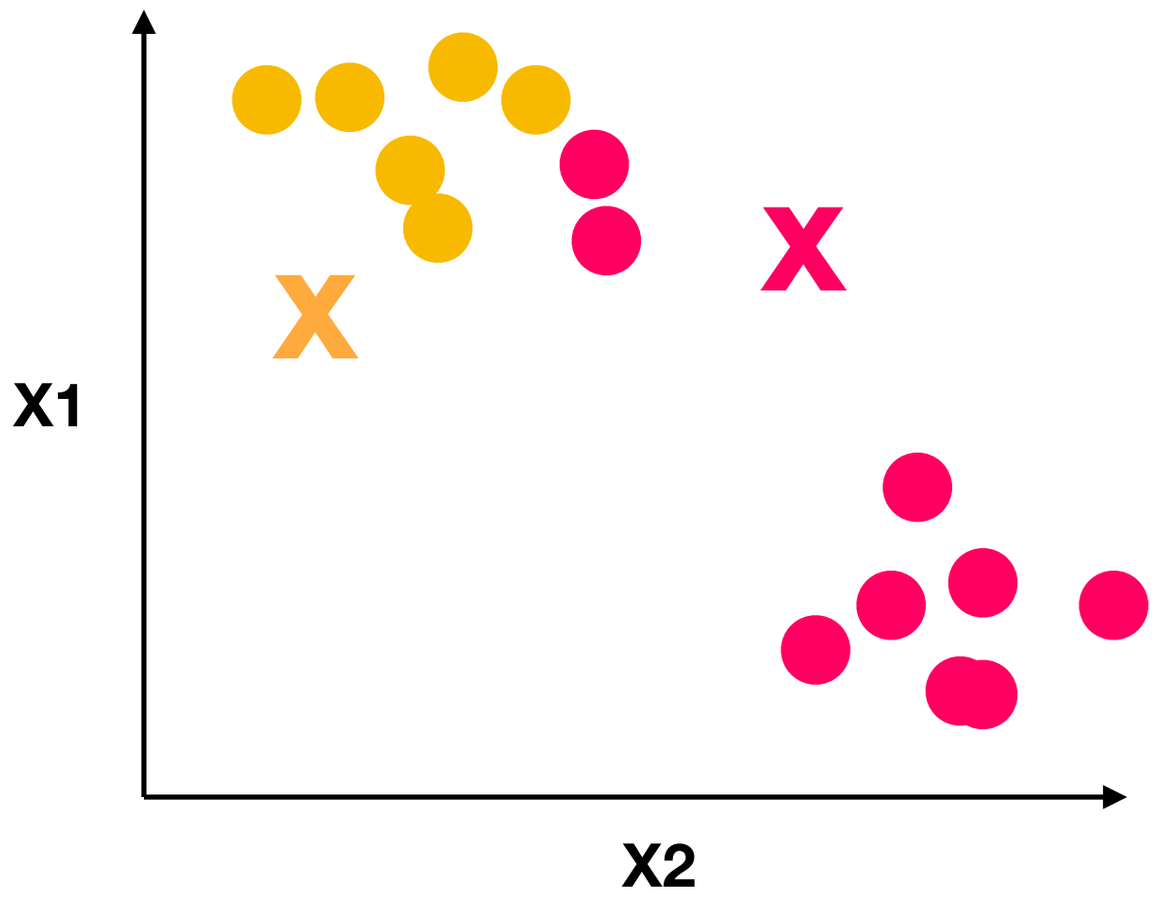
Step 1. Randomly Select K-Cluster "Centers"



CHIDATA

K-Means Algorithm

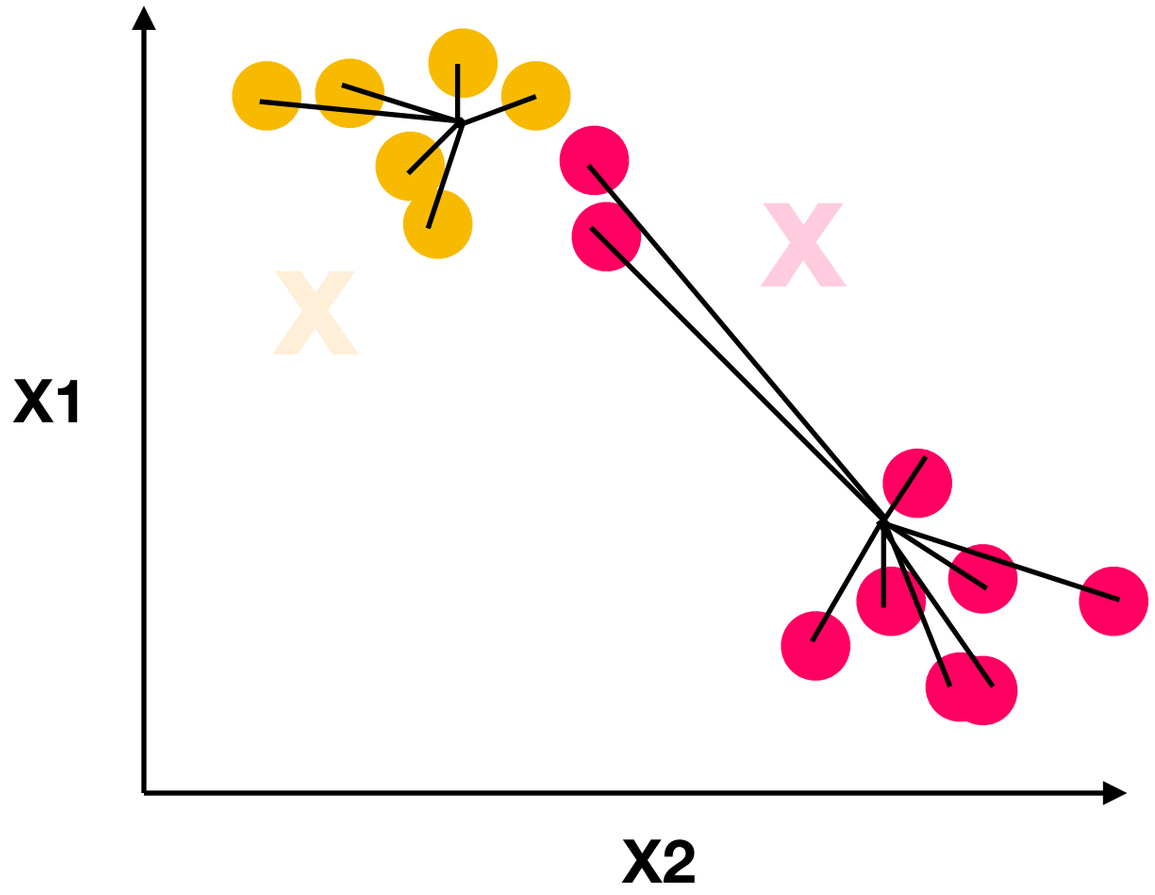
Step 2. Assign Each Example To Nearest Center



CHIDATA

K-Means Algorithm

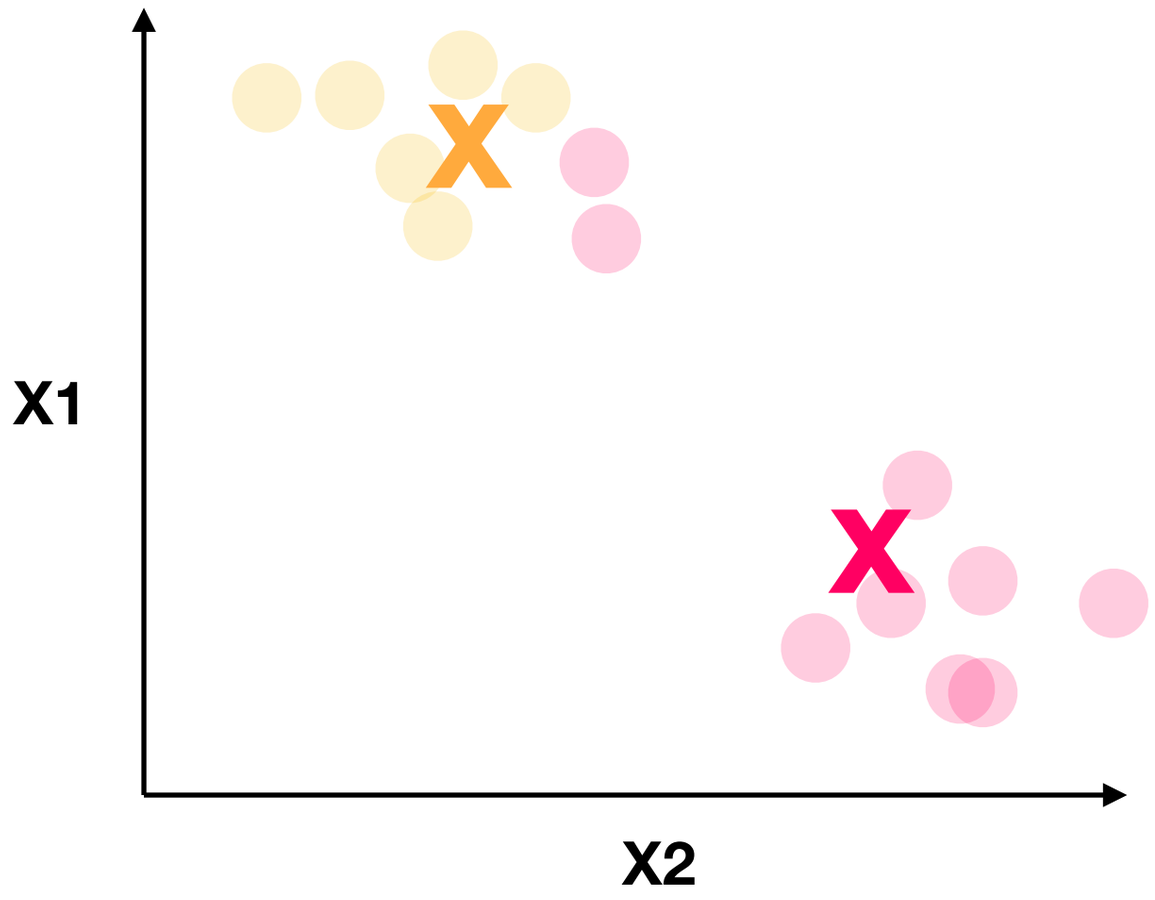
Step 3. Calculate the "Centroid" For Each Group of Data Points



CHIDATA

K-Means Algorithm

Step 4. The New Centers Are the Calculated Centroids

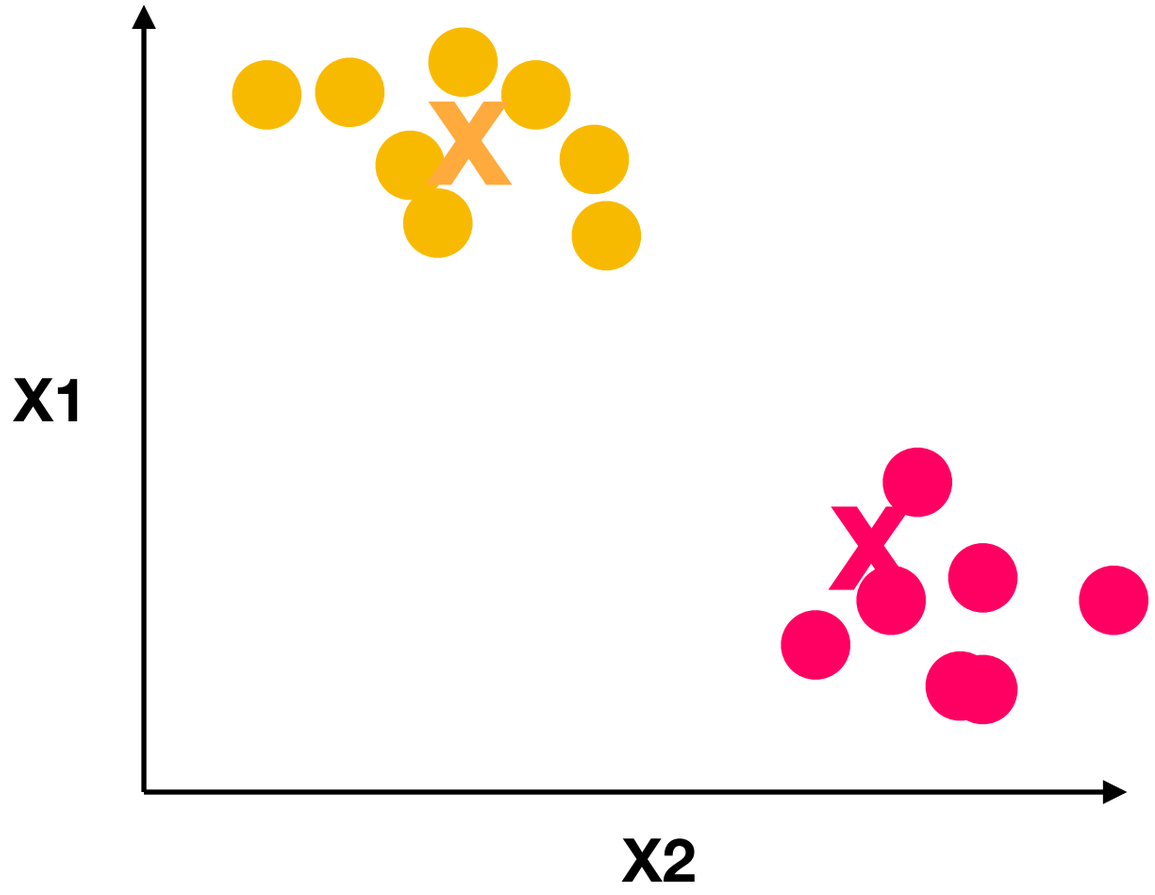


CHIDATA

K-Means Algorithm

Repeat Step 2. Assign Each Example To Nearest Center

Until Centers Stop Changing…



CHIDATA

PCA/Kmeans

No “labels” or “y’s”?

Class of models called “**generative**” models

Aside: Conditional Probability

Suppose we have two variables: **X=Grass_is_wet?** **Y=Was_raining?**

Joint probability distribution

P(X,Y)	Rain	No Rain	P(X)
Wet	0.20	0.2	0.4
Not Wet	0.10	0.5	0.6
P(Y)	0.3	0.7	

What is the probability of X and Y

Aside: Conditional Probability

Suppose we have two variables: **X=Grass_is_wet?** **Y=Was_raining?**

Conditional probability distribution

P(Y | X)	Rain	No Rain	P(X)
X=Wet	0.2/0.4=1/2	0.2/0.4=1/2	0.4
X=Not Wet	0.1/0.6=1/6	0.5/0.6=5/6	0.6

Given X what is the probability of Y

Aside: Conditional Probability

Suppose we have two variables: **X=Grass_is_wet?** **Y=Was_raining?**

Conditional probability distribution

P(Y | X)	Rain	No Rain
X=Wet	1/2	1/2
X=Not Wet	1/6	5/6

P(X)
0.4
0.6

If you don't know the original probabilities of X, you lose information.

Aside: Conditional Probability

generative

Cares about "structure" of X

P(X,Y)	Rain	No Rain
Wet	0.20	0.2
Not Wet	0.10	0.5

KMeans/PCA

discriminative

Ignores structure of X

P(Y | X)	Rain	No Rain
X=Wet	1/2	1/2
X=Not Wet	1/6	5/6

Linear Regression

Logistic Regression

Sanjay's 0.02

discriminative

Ignores structure of X

Vast majority of AI applications that actually "work" are discriminative models

P(Y | X)	Rain	No Rain
X=Wet	1/2	1/2
X=Not Wet	1/6	5/6

Sanjay's 0.02

generative

Cares about "structure" of X

P(X,Y)	Rain	No Rain
Wet	0.20	0.2
Not Wet	0.10	0.5

Generative models are very useful in scientific applications where structure matters.