Feature engineering là dùng những kiến thức về xử lí data để từ tập dữ liệu raw mình tạo ra được những features đặc trưng cho raw data đó để apply model máy học

[{"id": 1463, "name": "culture clash"}, {"id": 2964, "name": "future"}, {"id": 3386, "name": "space war"}, {"id": 3388, "name": "space colony"}, {"id": 3679, "name": "society"}, {"id": 3801, "name": "space travel"}, {"id": 9685, "name": "futuristic"}, {"id": 9840, "name":

* id: 1463
* name: "culture clash"

Data transformation: xữ lý categorical thành one hot, tìm cách để feature apply model

word embedding

Remove nếu data ích ảnh hưởng

Replace bằng 0

Replace bằng mean

Apply model

Nếu attribute là 1 đoạn văn, chuyển đoạn căn sang categorical, sử dụng **topic modelling**: extract keyword trong đoạn văn

**Sentiment analysis**: áp dụng khi cty collect comment của user, họ react ntn với iphone X

3 câu hỏi trong analytic report

First, data clean: remove missing value, if revenue < 0, remove, if revenue is not logic, revenue = 1e6

feature engineering: extract data

data exploration: correlation features, t-test, F-test. How?: what genres impact revenue, correlation, ttest between revenue, average score, if ttest pvalue > 0.05, 2 feature equal, one of 1 feature is not representative

Correlation between category and continuous:

1. 1

123 123

123 5465

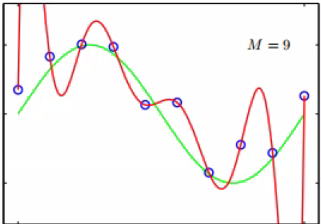
74 120

Run ANOVA

Correlation between **Name of movie and revenue**: **TF-idf**(term frequency: count # of word in this doc, inverse document frequency): count # of word between docs. more high word tf idf point, more representative this word is. Trade-off: don’t put context. Eg: `A loves B` equal `B loves A`

A/A test: put one model in Zing and NCT

**Central limit theorem**: if sample > 30, this variable normal dist. When released and unreleased not equal

Overfit: 

Run on train -> right

Run on test -> wrong

Bias-Varience Tradeoff

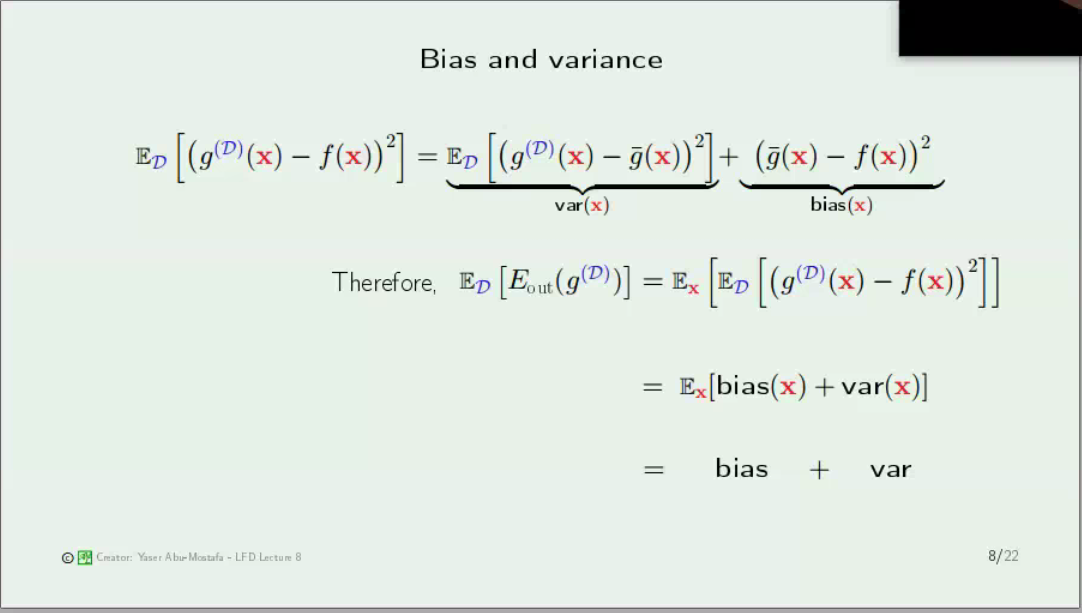
Hàm:

f: hàm lý thuyết (population)

g: hàm model học tốt nhất (on sample), what you get

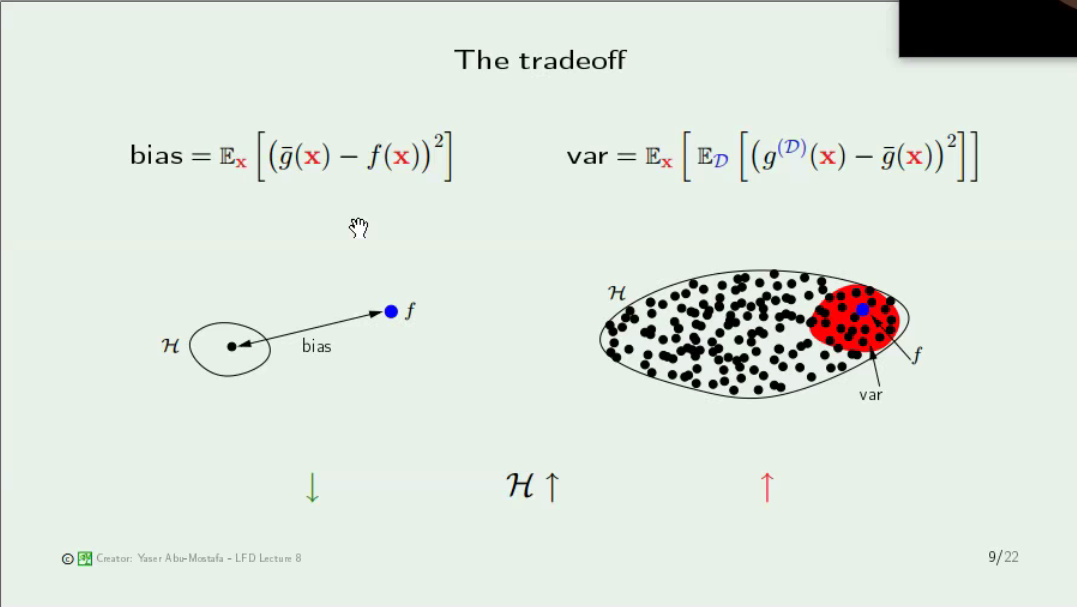
g trung bình(on population, never reach it): best for 100 sample, do for 100 sample to make g become g mean

variance: distance between g and g mean. How representative g is. If D is representative for population, variance = 0. More feature, more representative



bias: (find model ax+b, logx + b, e to make distance small). Bias high -> proba that wrong function is high

tradeoff bias variance:



H: set of function: ax+b, log(x), ..

pick random function from H

Bias = calculate distance between H and f. Bias high -> avoid by changing function

If variance large, proba that f in this set is large. if variance 0, increase attribute, more H need to calculate, the bias larger

PCA

reduce dimension: LDA, PCA

**Eigen-decomposition example**:









With  , you have 

With  , you have 



With  , you have 



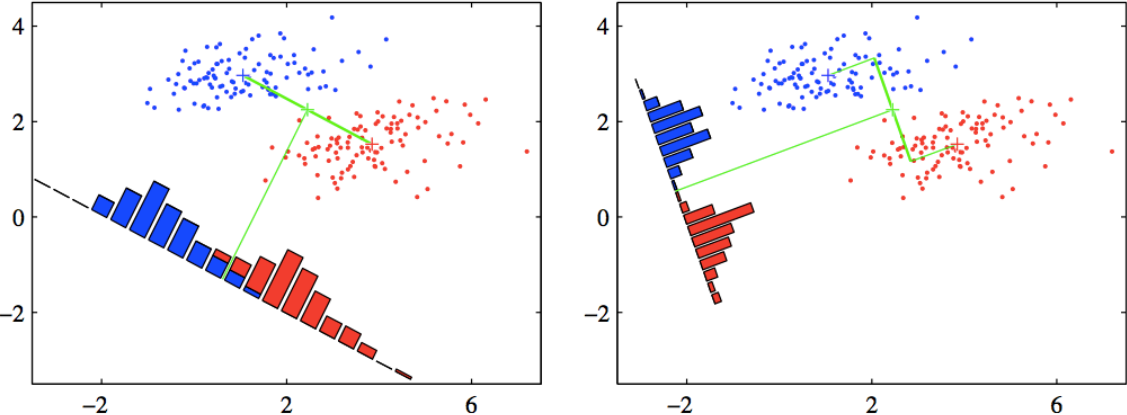
Eigen-decomposition



**Linear Discriminant Analysis**

There are supervised dataset: n class. How to reduce dimension of this dataset.

* Project on axis such that the variance of dataset is remained



The first one: red and blue is overlapped, so it is not optimized

The second one: red and blue is not overlapped, so it may be optimized

*Two constraints for LDA:*

Problem: N1 points of class C1 has mean m1 and N2 points of class C2 has mean m2

* The **between-class distance** is maximized



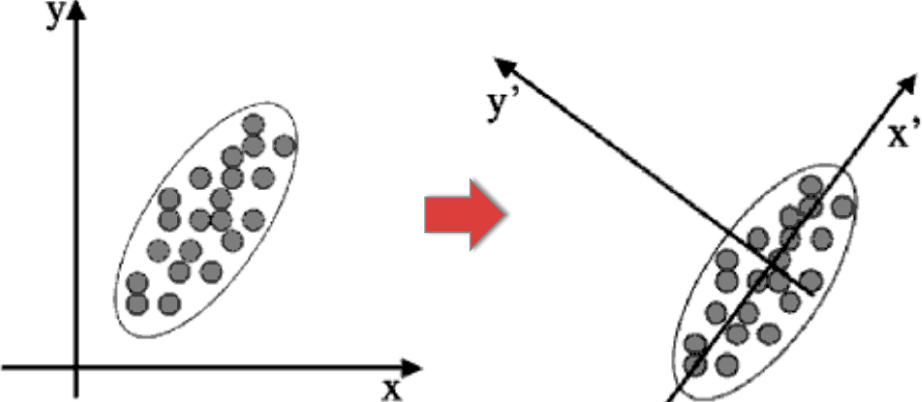
* The **within-class variance** is minimized



**Principal Component Analysis**

There are unsupervised dataset. How to reduce dimension of this dataset

* Project on axis such that the variance of dataset is remained



Mean Vector: 

Variance of projected data on dimension u1:



Maximized: 







With  , you have 



With  , you have 

With  , you have 





