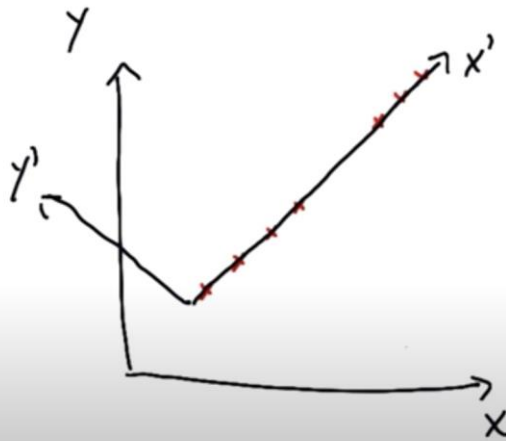


PRINCIPAL COMPONENT ANALYSIS - PCA

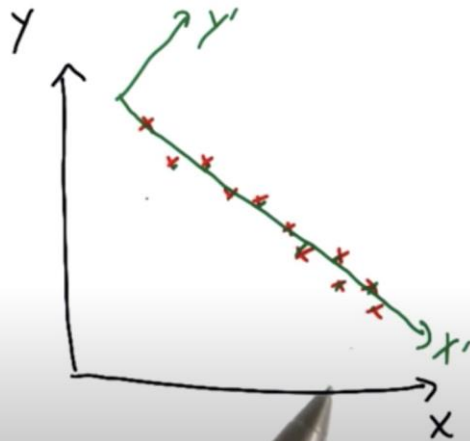


WHAT IS
DIMENSIONALITY
OF DATA

☒ 1
☐ 2

It's a 1 dimensional, if we rotate x & y axis by some degree

PRINCIPAL COMPONENT ANALYSIS - PCA

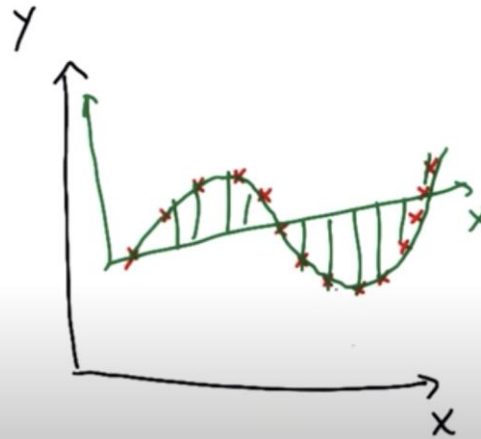


WHAT IS
DIMENSIONALITY
OF DATA

☒ 1
☐ 2

It's also 1D data as we can apply rotation and make it 1D as we want (Note variation in Y axis is minor so ignored)

PRINCIPAL COMPONENT ANALYSIS - PCA

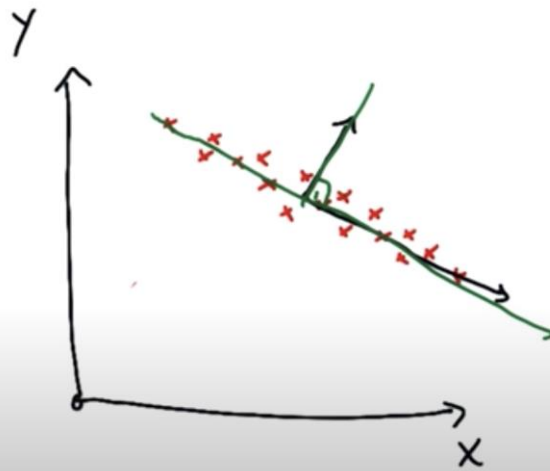


WHAT IS
DIMENSIONALITY
OF DATA

- 1
- ✗ 2

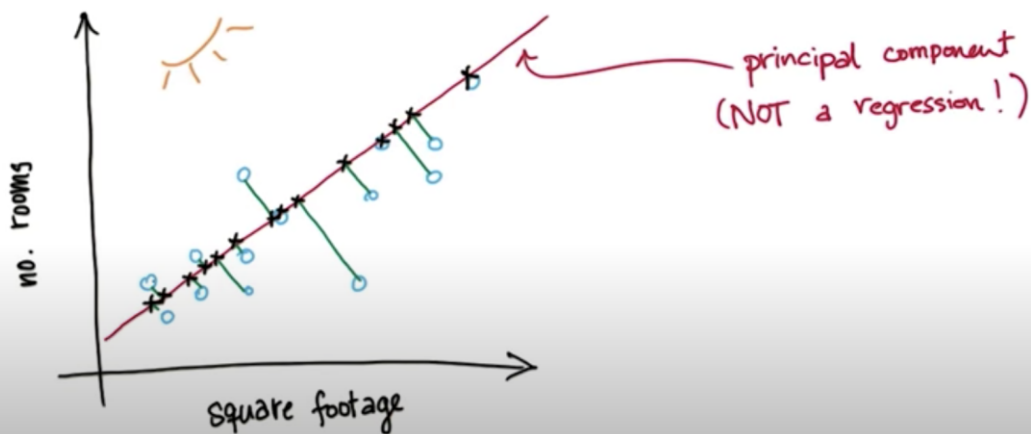
Still significant variation in Y axis after transformations, so 2 Dimension considered

PRINCIPAL COMPONENT ANALYSIS - PCA

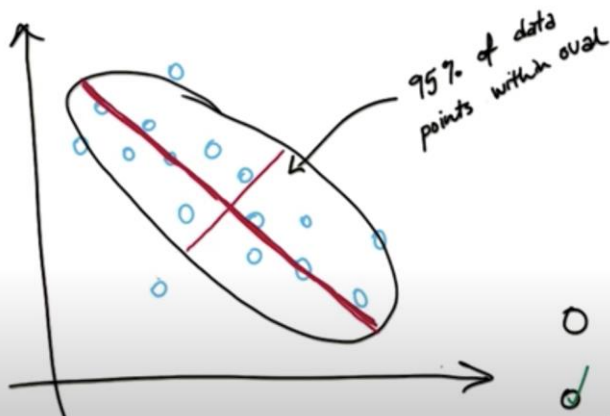


PCA -> Projection Idea (visualise)

Example: Square Footage + No. Rooms \rightarrow Size



How To Determine the Principal Component

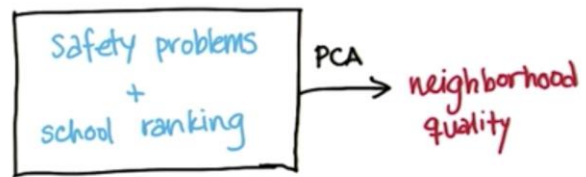
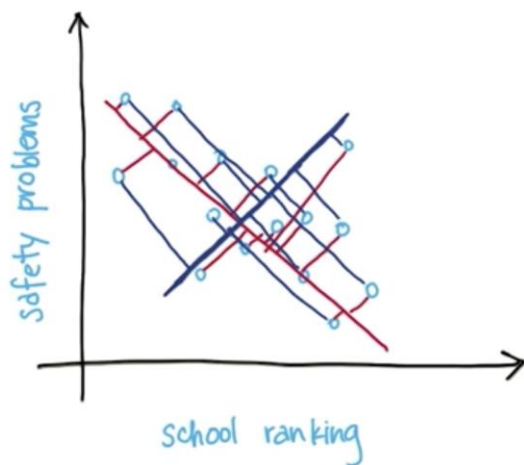


principal component of a dataset is the direction that has the largest variance because

- ☐ computationally easiest
- ☒ retains maximum amount of 'information' in original data
- ☐ just a convention

Mathematical Intuition (Information Loss minimal)

Maximal Variance and Information Loss



projection onto direction of maximal variance minimizes distance from old (higher-dimensional) data point to its new transformed value
→ minimizes information loss

GIST

Review/Definition of PCA

- systematized way to transform input features into principal components
- use principal components as new features
- PCs are directions in data that maximize variance (minimize information loss) when you project/compress down onto them
- more variance of data along a PC, higher that PC is ranked
- most variance / most information → first PC
second-most variance (without overlapping w/ first PC) → second PC
- max no. of PCs = no. of input features

When to use PCA ?

When To Use PCA

- latent features driving the patterns in data (big shots @ Enron)
- dimensionality reduction
 - visualize high-dimensional data
 - reduce noise
 - make other algorithms (regression, classification) work better b/c fewer inputs (eigenfaces)

Application Example

PCA for Facial Recognition

What makes facial recognition in pictures good for PCA?

- ☒ pictures of faces generally have high input dimensionality (many pixels)
- ☒ faces have general patterns that could be captured in smaller number of dimensions (two eyes on top, mouth/chin on bottom, etc.)
- ☐ facial recognition is simple using machine learning (humans do it easily)

IMP -> In a multiclass classification problem (more than 2 labels to apply), accuracy is a less-intuitive metric than in the 2-class case. Instead, a popular metric is the F1 score.

* Selecting number of Principal Components

Selecting A Number of Principal Components

Quiz: What's a good way to figure out how many PCs to use?

- just take top 10%
- ✓ train on different number of PCs, and see how accuracy responds — cut off when it becomes apparent that adding more PCs doesn't buy you much more discrimination
- perform feature selection on input features. before putting them into PCA, then use as many PCs as you have input features

REMEMBER :-

For PCA you will call fit() on training data, because you want to find pattern from training data