PCA :
(Principal Component Analysis).

It is a process of dimensionality Reduction.
let's say I have 500 features.
I train the model with initially 3 features.
Then I train with 10 features
Then I will train with 20 features.
All this time, I get a high accuracy.
lets say now, I train with 100 or 250 features, this time I get low accuracy.
This is called curse of dimensionality.

> As the no. of dimensions / features increase, there are chances that the model will perform poor.

This can be avoided using

1. feature selection
2. PCA / Dimensionality Reduction:
   - It is also called as feature extraction.
     The new features are derived from essence of existing features.

Why Dimensionality Reduction?

1. To prevent the curse of dimensionality.
2. To improve the accuracy of the model.
3. We can visualize or understand the data.

Why?

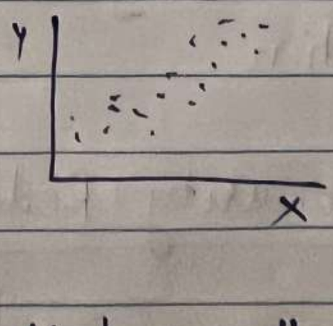
Human Being can visualize data upto 3D. we cannot visualize something like 100 dimension. We need to reduce it to 2D / 3D dimensions.

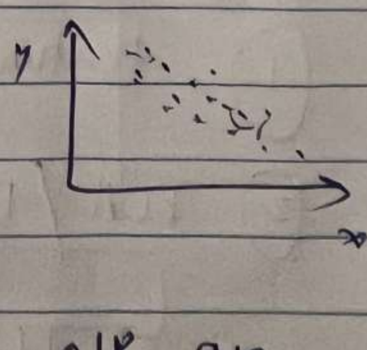
I/P	O/P
X	Y
-	-
-	-
-	-

There might be 2 scenarios

X ↑	Y ↑
X ↓	Y ↓

X ↑	Y ↓
X ↓	Y ↑





we can check how the I/P & O/P are correlated using covariance or coefficient.

spearman Rank.

$$\text{cov}(x,y) = \sum_{i=1}^{n} \frac{(x_i - \bar{x})(y_i - \bar{y})}{N-1}$$

Here the covariance is +ve or -ve.
But there is no specific range for covariance.
So we can use

$$\text{coefficient} = \frac{\text{cov}(x,y)}{\sigma_x \sigma_y}$$

It ranges between -1 to 1.
If a feature is highly positively correlated, it is closer to 1.
If a feature is negatively highly correlated, it is closer to -1.
If ~~is is~~ it is 0, they are not correlated.

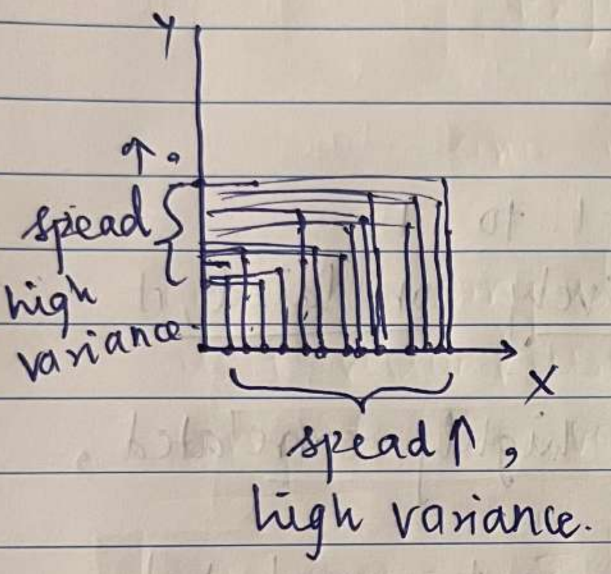
We can drop a feature that is not highly correlated [depends on domain expertise].

Feature Extraction:

We can derive or extract a new feature from the existing features using transformations.
This derived feature will also be able to predict the O/P.

PCA (Principal Component Analysis) :

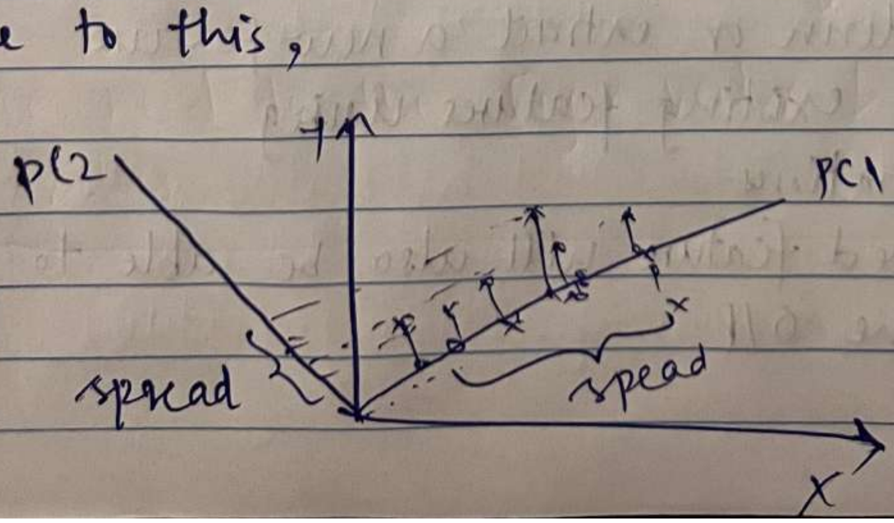
The main aim is to reduce dimensions. lets say I have 2D dimension, I need to reduce it to 1D dimension.



we can ~~d~~ reduce by projecting the data points on X axis. ~~T~~therefore we are ignoring the Y axis. Due ~~this~~ to this, a lot of information will get lost. (since the spread on Y axis is ↑).

This can be avoided using Eigen Decomposition on matrix.

Due to this,

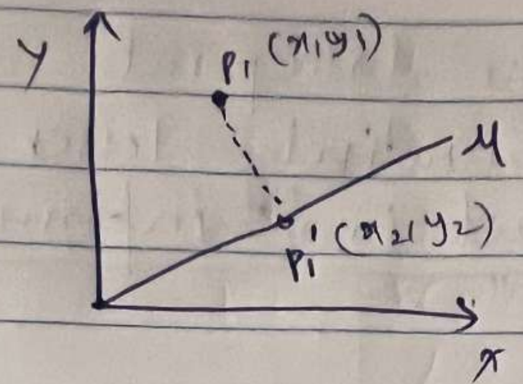


Now the spread for the data points along y axis is comparatively lesser. Therefore, less amount ~~in~~ of information will get lost.

Therefore, the final purpose ~~to~~ is to find the principal components with the maximum spreads (max. variance)

To convert 2D to 1D
PC1, PC2 to be constructed
$\underbrace{var(PC1)}_{\text{This is to be chosen.}} > var(PC2)$

To convert 3D to 2D
PC1, PC2, PC3
$\underbrace{var(PC1) > var(PC2)}_{\text{This is to be chosen.}} > var(PC3)$



we need to project the point onto the unit vector.

Projection formula $\frac{P_1 \cdot u}{||u||}$

since it is a unit vector, $||u|| = 1$.

once we compute all the points on unit vector (using projection), we will compute the variance.

$$\boxed{\text{variance} = \sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{N}}$$

This is the cost function.

we cannot do try and run and then find the best unit vector.

$\boxed{\text{We need to use Eigen vectors \& Eigen values.}}$

1. Calculate the covariance matrix. bet^n features.
2. With the matrix, determine the eigen value & eign vector.
3. ~~High~~ The highest Eigen value, the corresponding eigen vector needs to be selected.

$$\boxed{Av = \cancel{k}\cdot \lambda v} \rightarrow \text{linear transformation.}$$

A = covariance matrix.

let's say I have 2 features, ∴ It will be a 2×2 matrix.

$$\begin{array}{c c} & \begin{array}{cc} x & y \end{array} \\ \begin{array}{c} x \\ y \end{array} & \begin{bmatrix} var(x) & cov(x\,y) \\ cov(y\,x) & var(y) \end{bmatrix} \end{array}$$

If I have 3 features, it will be 3×3 matrix.

we need to apply linear transformation on vector 'v', we will get λ values
↓
Eigen value.

If for 2 features, I will get 2 λ values (λ_1, λ_2).

$$\begin{array}{cc} \lambda_1 & \lambda_2 \\ \downarrow & \downarrow \\ PC1\ ✓ & PC2 \end{array}$$

(1D) $\longrightarrow$ $\boxed{1D}$.

3D → 2D

λ_1 , λ_2 , λ_3

↓ ↓ ↓

PC1 PC2 PC3

$\boxed{1D} + \boxed{1D} \rightarrow 2D.$