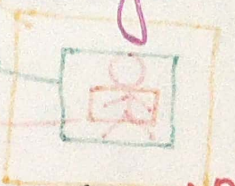
Principal Component Analysis (PCA)

Steps involved in ml model building:-



1. Data ingestion
2. EDA
3. preprocessing → feature engineering → Dimension Reduction
4. Model Building
5. Model evaluation.

- Feature selection
- Feature Extraction.

* Dimension - no. of columns or features
* Dimension reduction - process of Reducing the dimension or feature.

→ Before understanding FS and FE, we should know about curse of dimensionality.

How does the curse of dimensionality occur?

→ Let's say we have a dataset with mxn size.
(m = row, n = column.)

→ As we add more dimensions to our dataset, the volume of space increases exponentially. This means that the data becomes sparse. Think of it this way: if you have line (1D), it's easy to fill it with few points. If you have a square (2D), you need more points to cover the area. Now imagine a cube (3D) - you'd need even more points to fill the space. This concept extends to higher dimensions, making the data extremely sparse.

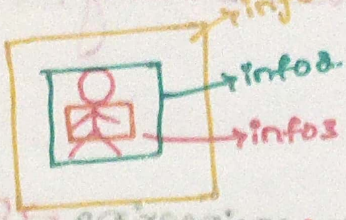
Ex:- Data (Row x column)

	Model	Accuracy	
f_1 ------ f_{10} y	→ M_1	75%	} → less features
f_1 ------ f_{100} y	→ M_2	80%	} → optimal (having sufficient features)
f_1 ------ f_{1000} y	→ M_3	72%	} → many

f_1 ------ f_{1000} = independent features; y = dependent feature

many → overfitting → sparsity (lot's of 0's and less 1's.)

Ex:- which information is precise for our model to identify a image



⇒ second info is very precise to capture the image.

→ what problems does it cause?

1. Data sparsity :- As mentioned data becomes sparse, meaning that most of the high-dimensional space is empty (in high dimensional space, most data points are at the "edges" or "corners" making data sparse). This makes clustering and classification tasks challenging.

2. Increased computation :- More dimensions mean more computational resources and time to process the data.

3. overfitting :- with Higher dimensions, models can become overly complex, fitting to the noise rather than the underlying pattern. This reduces the model's ability to generalize to new data.

4. Distance loose meaning :- In high dimensions, the differences in distances between data points tend to become negligible, making measures like euclidian distance less meaningful.

5. performance degradation :- Algorithms, especially those relying on distance measurements like K-nearest neighbours, can see a drop in performance.

6. visualization challenges :- visualization challenges High dimensional data is hard to visualize, making exploratory data analysis more difficult.

How to solve the curse of dimensionality :-

Two different ways are there to resolve the curse of dimensionality

1. Feature selection.
2. Feature extraction / Dimension Reduction / Feature transformation.

Feature selection:- Feature selection is a process that chooses a subset of features from original features so that the feature space is optimally reduced according to a certain criterion.

Ex:- $f_1, f_2, \ldots, f_{100} \Rightarrow$ Model

$f_1, f_2, \ldots, f_{100}$ ⇓ all features or not important

Model ⇓ 75%

we may select optimal no. of feature (i.e; 50 or 60 or 70 - -)

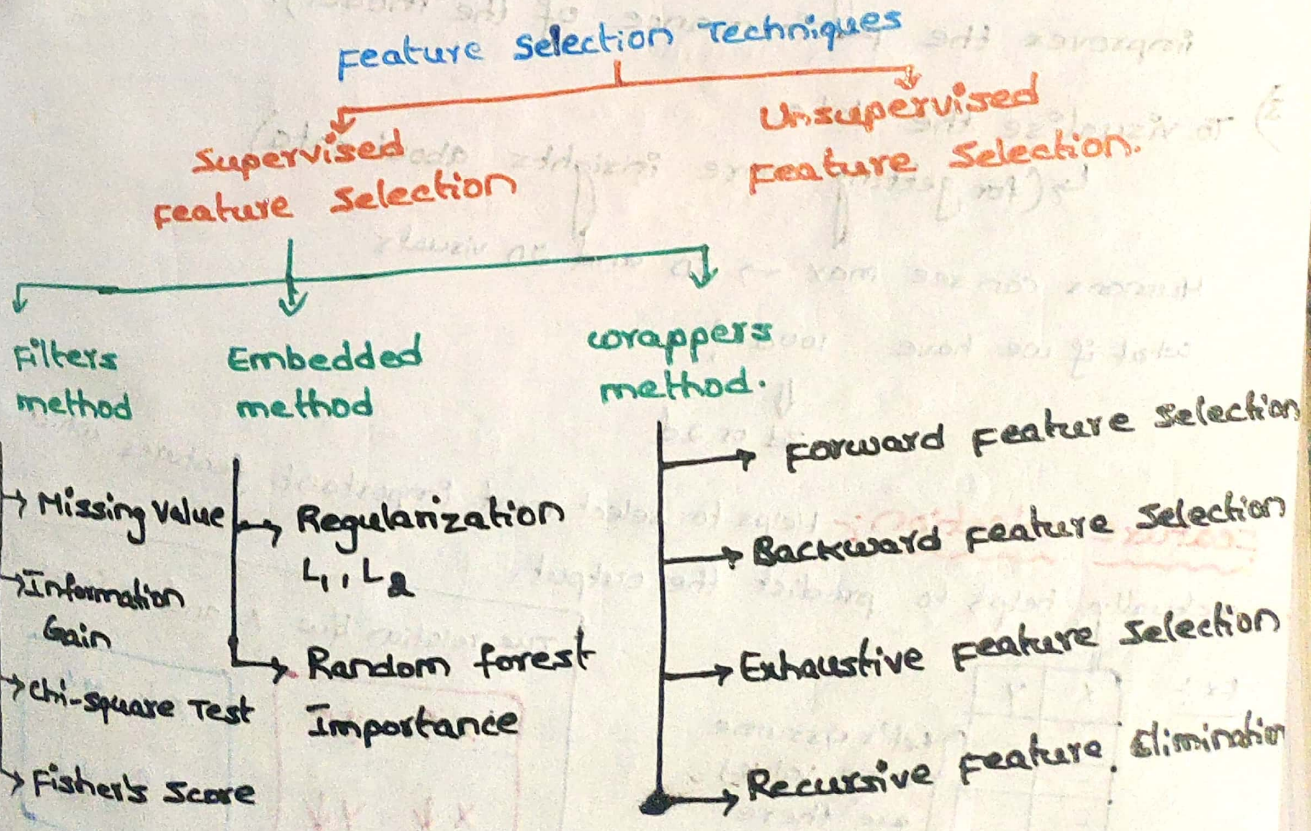
→ There are mainly two types of feature selection techniques;

1) **Supervised feature selection Technique:-**

→ Supervised feature selection techniques consider the target variable and can be used for the labelled dataset.

2) **unsupervised feature selection Technique:-**

→ unsupervised feature selection techniques ignore the target variable and can be used for unlabelled dataset.



2) Dimensionality reduction / Feature Extraction / Feature transformation

"Dimensionality Reduction is a type of feature extraction technique that aims to reduce the number of input features while retaining as much of the original information as possible.

Dimensionality Reduction technique :-

1. PCA → principal component Analysis.
2. t-SNE → t distributed stochastic Neighbour embedding.
3. LDA → Linear discriminate Analysis.

Why dimensionality Reduction?

1) prevent curse of dimensionality
2) To improve the performance of the model
   (To train all the features it takes more time, but if we do the dimensionality reduction it will take less time and improves the performance of the model).
3) To visualise the data
   ↳ (for getting more insights about data)

Humans can see max → 3D and 2D visuals

what if we have 100d?

⇓

3d or 2d

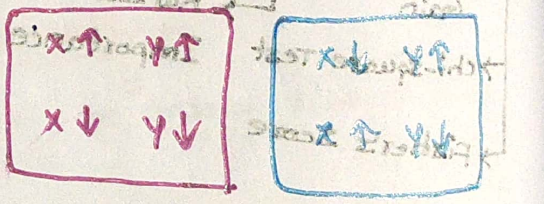
Feature selection :- Helps to select most important features which actually helps to predict the output.

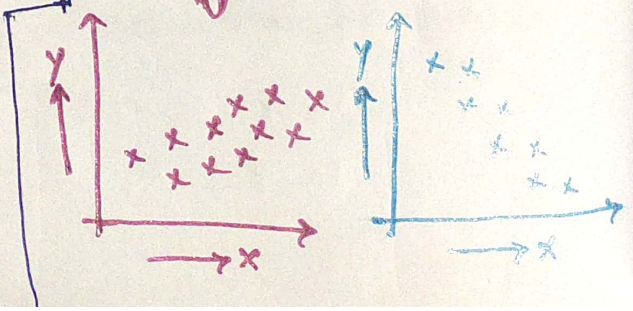
Ex :-

x	y
-	-
-	-
-	-
-	-

Let's Assume some values are there corresponding to x (I/p) and y (o/p) features).

True relation b/w x and y is as follows

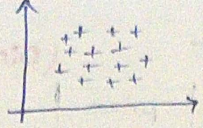




→ we can quantify the Relation interms of mathematical terms.

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

= +'ve
= -'ve
= 0

$cov(x,y) = 0 \Rightarrow$ 

→ whichever features have this kind of highly positive covariance, those features are super important.

X is important for predicting Y.

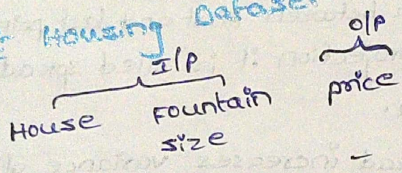
pearson correlation coefficient: $= \frac{cov(x,y)}{\sigma_x * \sigma_y} = -1$ to 1

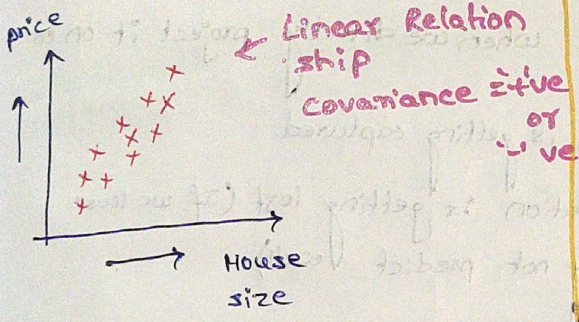
→ The more towards the value of +1, the more +'ve correlated X & Y is.

→ The more towards the value of -1, the more -'ve correlated it is.

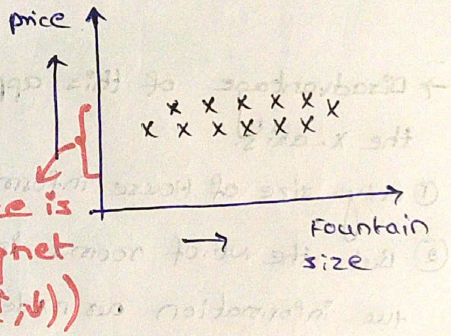
→ ≈ 0 {No relationship}

Ex: Housing Dataset.





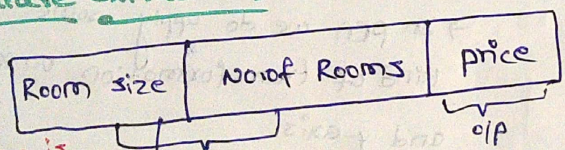
Covariance ≈ 0



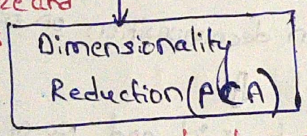
→ Due to '0' covariance we can conclude fountain size is not important and drop this feature.

feature extraction:-

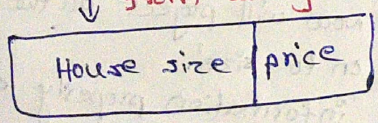
Ex:



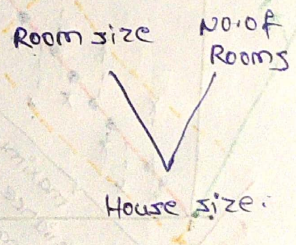
covariance is either +'ve or -'ve for room size and no.of rooms



→ extracts new feature "House size" from already there features.



2 feature → 1 feature



PCA Geometric Intuition:- (Dimensionality Reduction)

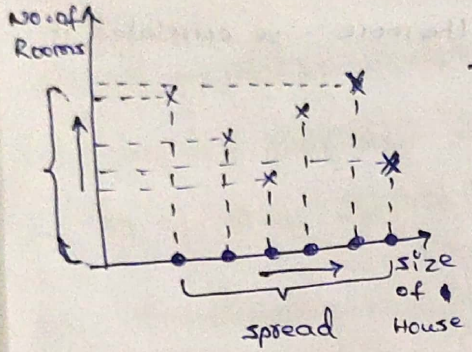
Size of House	No. of Room	price
		↓ o/p

with PCA

⇓

2 dimension → 1 dimension

→ let's plot the above I/p features in the form of scatter plot



Approach - 1

→ when we project all the data points on to the x-axis, we may get all the data points in one dimension.

→ The area between first and last point after projection it is called spread of the data.

→ If spread increases variance also increases.

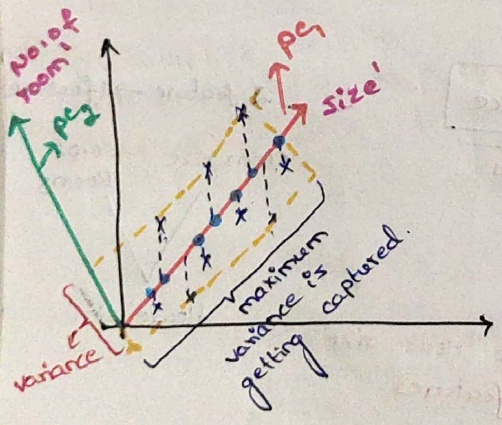
→ Disadvantage of this approach when we directly project it on to the x-axis.

① only size of House information is getting captured.

② But the No. of rooms information is getting lost (If we loose the information our model may not predict well)

↓

Here we are doing feature extraction with lot of information lost



Approach - 2

→ In PCA we do apply some kind of transformation on x-axis and y-axis.

⇓

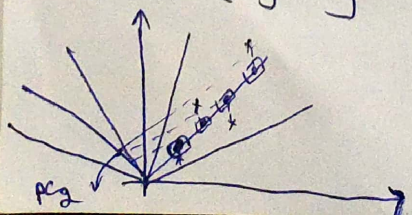
→ Eigen decomposition on matrix

⇓

→ we will get size' and No. of Room'

→ Now will project all the information on to size'.

→ Through size' we can capture the information properly compared to approach 1 (projecting on to the x-axis).



→ PCA algorithm finds 2 principal components, when we project the points on to the line whichever is able to capture more variance.

→ PC_1 will capture the maximum amount of variance.
→ PC_2 will capture the next maximum amount of variance after PC_1
→ If we have three dimensions, PC_3 will capture the next maximum amount of information after PC_1 and PC_2. (2D → 1D)
→ The Geometric Intuition behind PCA is to find out principal components i.e, we need to find out one line that should be able to capture maximum amount of variance.
→ once we take this PC_1, we can convert this ~~from~~ dataset of 2D from 1D.

Why variance is important?

→ variance talks about the spread of the data.
→ with the help of variance we can capture more information.

2 Dimension	3 Dimension
PC_1, PC_2	PC_1, PC_2, PC_3
$var(PC_1) > var(PC_2)$	$var(PC_1) > var(PC_2) > var(PC_3)$

Mathematical Intuition Behind PCA Algorithm.

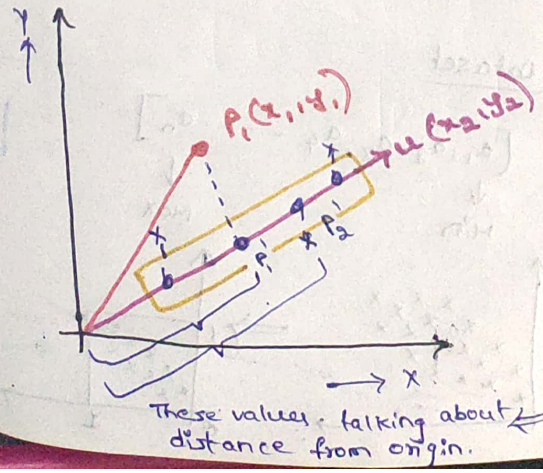
2D → 1D

Aim: To find best PCA line

For achieving this needs to consider two terms

① projection
② cost function (max-variance)

→ when we project points on to the line maximum variance is getting captured.
→ out of all these points let's take up one point.



These values talking about distance from origin.

projection of point P_i on unit vector u

$$proj_{P_i} \cdot u = \frac{P_i \cdot u}{\|u\|}$$

$\|u\| = 1 \Rightarrow$ unit vector

$$proj_{P_i} \cdot u = P_i \cdot u \Rightarrow \text{scaler value}$$

$P_0', P_1', P_2', P_3', P_4', \dots, P_n'$ → scaler values

$P_0', P_1', P_2', P_3', P_4', \ldots, P_n'$

$\Downarrow$

$x_0', x_1', x_2', x_3', x_4', \ldots, x_n'$

{Goal :- find the best unit vector which captures maximum variance)

$$\text{Max-variance} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n} \Rightarrow \text{cost Function.}$$

→ we should not go ahead and select different unit vectors to find out best vector.

→ In order to find that unit vector we have a method eigen decomposition

→ How do we find out which unit vector is best to capture maximum variance, so for that we will be using "Eigen values and Eigen vectors".

→ ① First we need to find covariance matrix between features

② Eigen vectors and Eigen values will find out from this covariance matrix

$$Av = \lambda v$$

③ whichever the eigen vector is the largest one, for that eigen vector, the "eigen value" is high.

$\Downarrow$

magnitude of eigen vector

(captures the maximum variance)

Steps involved in pca

① standardize the data
② covariance matrix b/w features.
③ Find out eigen value and eigen vector.
④ principal component.

scaling of the data :-

Min-Max standard scaler $\Rightarrow$ Suppress the data [0-1]

↳ Data point

$$\text{Min-Max} = \frac{X - \text{Min}}{\text{Max} - \text{Min}}$$

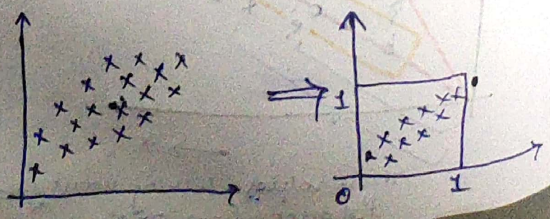
$$= \frac{\text{Min} - \text{Min}}{\text{Max} - \text{Min}} = 0$$

$$= \frac{\text{max} - \text{min}}{\text{max} - \text{min}} = 1$$

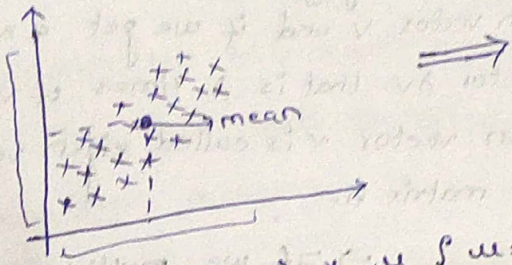
Dataset

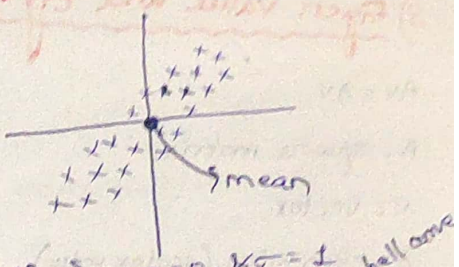
$[a_1, a_2, a_3, a_4 \ldots a_n]$

↓ Min ↓ Max

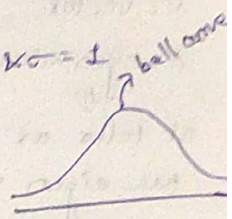


1) Standardization of the Data :





SND ⇒ mean = 0, σ = 1



standardization ⇒ $\frac{x-\mu}{\sigma}$ { μ = mean, σ = std

Data → standardization → standard data ($\mu = 0, \sigma = 1$)

Variance

2, 8, 10, 12, 6, 5, 13, 14

↓ mean

↓ deviation ⇒ (mean − dp)^2 (dp = data point)

↓ mean of deviation.

$\Rightarrow \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2$

↓

variance ⇒ spread of the data around the mean.

2) covariance matrix of x, y :

i/p → x, y → o/p

x → f_1, y → f_2

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

$$cov(x,x) = \frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})^2 = var(x)$$

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

$$= \frac{1}{N}\sum_{i=1}^{N}(y_i-\bar{y})^2 = var(y)$$

$$cov(x,y) = cov(y,x)$$

For 2D

	x	y
x	cov(x,x)	cov(x,y)
y	cov(y,x)	cov(y,y)

⇒

var(x)	cov(x,y)
cov(y,x)	var(y)

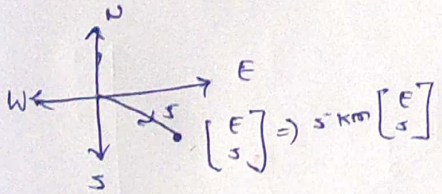
↔ covariance matrix of x, y features

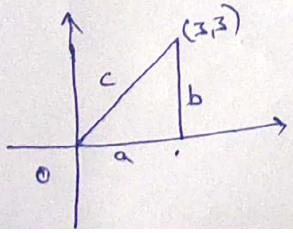
$\Rightarrow \frac{1}{N}\sum_{i=1}^{N}(x_i-\bar{x})(y_i-\bar{y})$ → std = 1, mean = 0

$\Rightarrow \frac{1}{N}\sum_{i=1}^{N} x_i y_i$

$x_1y_1 + x_2y_2 + x_3y_3 + \dots + x_4y_4$

↳ dot product of two vectors





$c^2 = a^2 + b^2$

$c = \sqrt{a^2+b^2}$

$c = \sqrt{3^2+3^2}$

$= \sqrt{18} = 4.2$

$OC = 4.2 \begin{bmatrix} 3 \\ 3 \end{bmatrix}$

$= 3\hat{i} + 3\hat{j}$

3) Eigen value and Eigen vector

$Av = \lambda v$

A = square matrix

v = vector

λ = eigen value (scalar value)

↓

It tells us how much the eigen vector changes in size when we multiply with matrix.

→ If we multiplying a square matrix A with vector v and if we get a new vector λv that is λ times of vector v then vector v is called eigen vector of matrix A.

→ If we multiplying matrix A by vector v the new vector λv does not change the direction.

$Av - \lambda v = 0$

$(A - \lambda I)v = 0$

$\text{Det}|A - \lambda I|$ ⇒ I = identity matrix.

λ_1, λ_2 ⇒ eigen values.

↓ ↓

PC_1 PC_2

3D → 2D

$\lambda_1, \lambda_2, \lambda_3$

↓ ↓ ↓

PC_1 PC_2 PC_3

1D ⇒ 2D

3D → 1D

$\lambda_1, \lambda_2, \lambda_3$

1D

2D → 1D

λ_1, λ_2

↓ ↓

PC_1 PC_2

1D