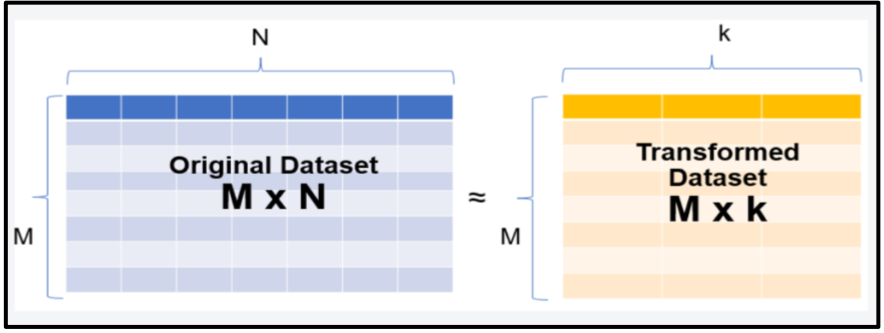
# PCA-Machine Learning Concept

## PCA

- It stands for Principal Component Analysis.
- It is a procedure to convert corelated variables into new components such that these components are un corelated to each other and there is no multicollinearity.
- We try and convert large datasets into smaller ones that has fewer variables.
- This improves model performance

## Problem

- There are lot of variables to visualize and explore.
- In the figure below our original data set has **M rows and N columns** for EDA.



## Solution

- Once we have applied PCA by specifying the no. of components we get a transformed data set with **M*k** features

## Math behind PCA

- **Vectorial Representation of data**
- If we have data set as below:

| Patient ID | Height (cm) | Weight (kg) |
|---|---|---|
| P1 | 165 | 55 |
| P2 | 155 | 71 |

- We will create a matrix out of this i.e., [165,55] and represent this as a vector.
- Vector representation will be (165,55) , $[165,55]^{T}$ (transpose) etc.
- It can also be written as **165i+55j**
- We calculate the magnitude using the Pythagoras theorem.
- In vector addition we add the ith terms together.

## Basis Vector

- We find the basis vector which is along the best fit line that maximizes the variance. That will be **PC1.**
- Next is to find the vector is which is perpendicular to that component. This will be **PC2.**
- If there were 3 dimension's we would have found PC3 as we found PC1 and PC2. It will be perpendicular 1st and 2nd principal component.
- Original dataset →PCA basis

- The number of principal component are same as no. of columns.
- The algorithm by which PCA maximizes the variance is by eigen decomposition of the covariance matrix.

Applying PCA

- StandardScaler function. we did x=scaler.fit_transform(x) to have values in the same range
- We import PCA using from sklearn.decomposition import PCA and then do the fit.
- pca.components_ gives us the basis vectors
- pca.explained_variance_ratio_ gives us the amount of variance explained by each component it is same as the no. of attributes.

Scree Plot

- The plot gives us the amount of variance of each component on the Y-axis whereas X-axis is the no. of components.
- We can decide how many components we need as per this.

**We can then use the obtained features and perform other ML techniques to improve model performance.**