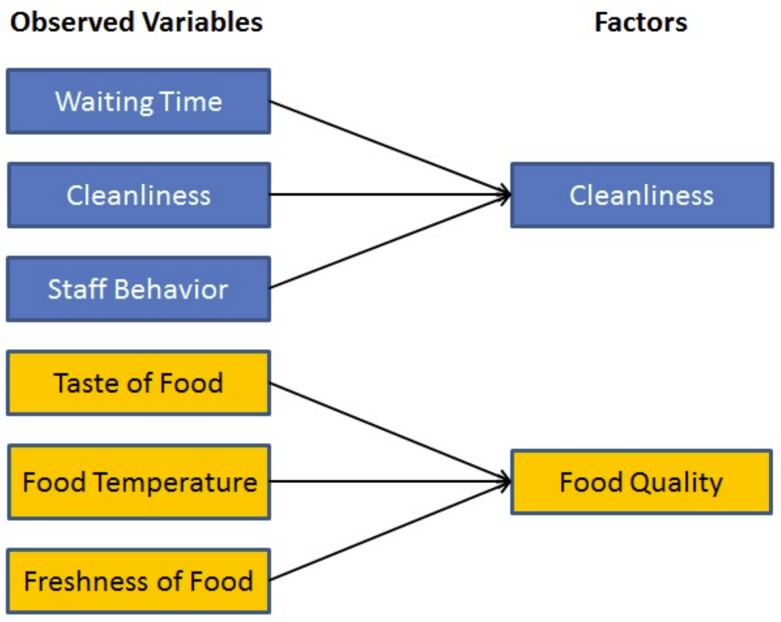
Feature Extraction

- Feature Extraction is the technique where you extract new features using the existing one.
- In this module, we will look at some prominent Factor Based feature extraction techniques based on along with its implementation in python.

1. Factor Analysis

- Factor Analysis (FA) is an exploratory data analysis method used to search influential underlying factors or latent variables from a set of observed variables.
- It helps in data interpretations by reducing the number of variables. It extracts maximum common variance from all variables and puts them into a common score.
- It is used basically in market research, advertising, psychology, finance, and operation research.



When We use this ??

- There are no outliers in data.
- The sample size should be greater than the factor.
- There should not be perfect multicollinearity.
- There should not be homoscedasticity between the variables.

Types of factor analysis:

1. Exploratory factor analysis (EFA)
2. Confirmatory factor analysis (CFA)
3. Multiple Factor Analysis
4. Generalized Procrustes Analysis (GPA)

To study more follow this https://www.geeksforgeeks.org/introduction-to-factor-analytics/ (https://www.geeksforgeeks.org/introduction-to-factor-analytics/)

Implementation

In [1]:
```
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris
from sklearn.decomposition import FactorAnalysis
```

In [2]:
```
iris = load_iris()
X=iris.data
variable = iris.feature_names
```

In [3]:
```
factor = FactorAnalysis(random_state=0)
factor.fit_transform(X)
pd.DataFrame(factor.components_,columns = variable)
```

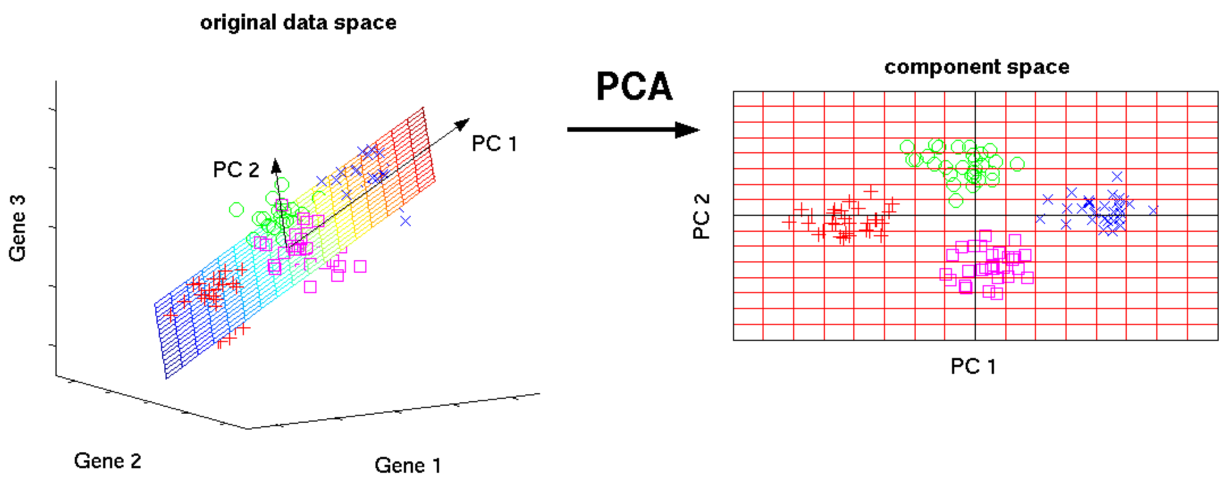
Out[3]:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	0.706989	-0.158005	1.654236	0.70085
1	0.115161	0.159635	-0.044321	-0.01403
2	-0.000000	0.000000	0.000000	0.00000
3	-0.000000	0.000000	0.000000	-0.00000

- In this way factor analysis occured

2. PCA (Principal Component Analysis)

- The Principal Component Analysis(PCA) is a way of reducing the dimensions of a given dataset by extracting new features from the original features present in the dataset. So it combines our input variables(or features) in a specific way and gives “new” features by retaining the most valuable information of all the original features. The “new” variables after PCA are all independent of one another.



Why it is used :

- It is used when we need to tackle the curse of dimensionality among data with linear relationships, i.e. we're having too many dimensions (features) in your data causes noise and difficulties (it can be sound, picture or context). This specifically get worst when features have different scales (e.g. weight, length, area, speed, power, temperature, volume, time, cell number, etc.)

When should I use PCA:

1. Do you want to reduce the number of variables, but aren't able to identify variables to completely remove from consideration?
2. Do you want to ensure your variables are independent of one another?
3. Are you comfortable making your independent variables less interpretable?

If you answered "yes" to all three questions, then PCA is a good method to use. If you answered "no" to question 3, you should not use PCA.

Point to remember while doing PCA:

- The first step we need to do while performing PCA is data standardization. i.e. scaling the data to mean as 0 and variance as 1.

Implement

In [4]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
from sklearn.datasets import load_breast_cancer
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
```

In [5]:
```
#loads breast cancer dataset into variable by name cancer.
cancer = load_breast_cancer()
# creating dataframe
df = pd.DataFrame(cancer['data'], columns = cancer['feature_names'])
# checking head of dataframe
df.head()
```

Out[5]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	d
0	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	
1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	
2	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	
3	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	
4	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	

5 rows × 30 columns

In [6]:
```
df.shape
```

Out[6]:
```
(569, 30)
```

- So clearly, we can see that there are 30 columns and each of them are numerical values. So we can directly apply PCA.

- before applying PCA we need to scale or feature

In [7]:
```
scalar = StandardScaler()
  # Standardizing
scalar.fit(df)
scaled_data = scalar.transform(df)
```

In [8]:
```
# applying PCA
pca = PCA(n_components = 3)
pca.fit(scaled_data)
x_pca = pca.transform(scaled_data)
```

- In this, what we are doing is standardizing the data(i.e. df) and applying PCA on it.
- There, n-components represents the number of principal components(i.e. new features) that we want to.

```
In [9]: print("Befor PCA shape: ",df.shape)
        print("After PCA shape: ",x_pca.shape)
```

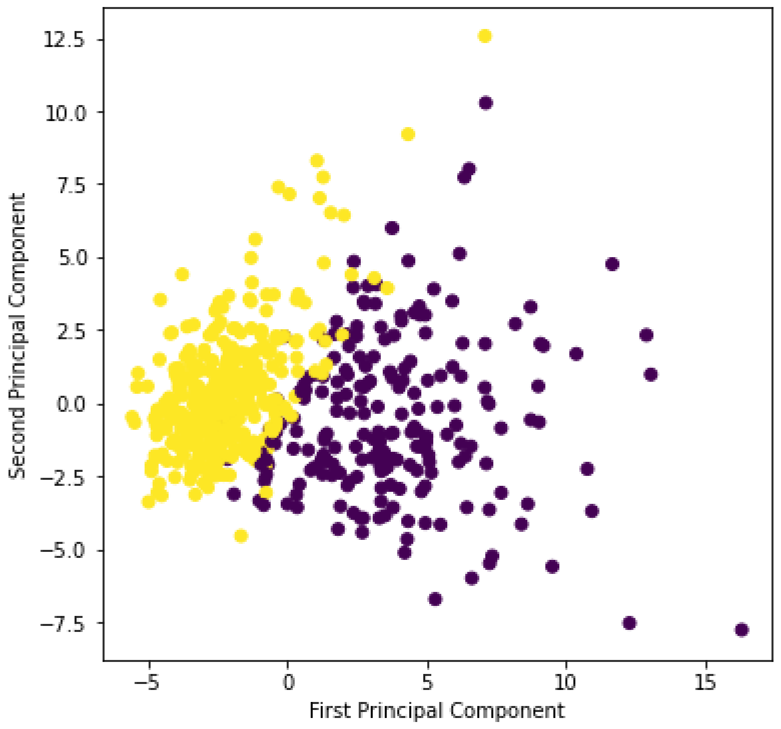
```
Befor PCA shape:  (569, 30)
After PCA shape:  (569, 3)
```

```
In [10]: plt.figure(figsize =(6,6))

         plt.scatter(x_pca[:, 0], x_pca[:, 1], c = cancer['target'])

         # labeling x and y axes
         plt.xlabel('First Principal Component')
         plt.ylabel('Second Principal Component')
```

```
Out[10]: Text(0, 0.5, 'Second Principal Component')
```

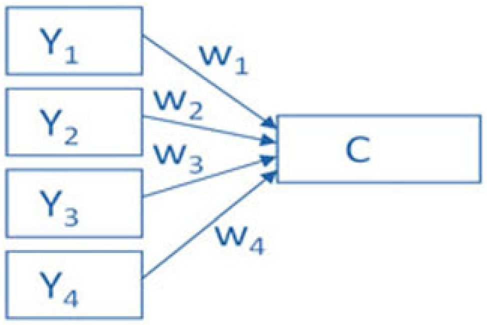


- In this step, I'm just displaying the 2-Dimensional plot of the data and it looks something like this.

Factor Analysis Vs. Principle Component Analysis

- PCA components explain the maximum amount of variance while factor analysis explains the covariance in data.
- PCA components are fully orthogonal to each other whereas factor analysis does not require factors to be orthogonal.
- PCA components are uninterpretable. In FA, underlying factors are labelable and interpretable.
- PCA is a kind of dimensionality reduction method whereas factor analysis is the latent variable method.
- PCA is a type of factor analysis. PCA is observational whereas FA is a modeling technique.

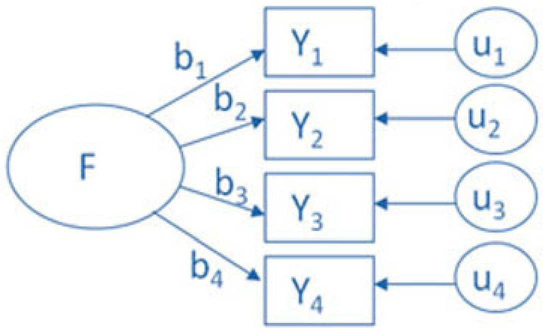
PCA



This model can be set up as a simple equation:

C = w1(Y1) + w2(Y2) + w3(Y3) + w4(Y4)

Factor Analysis



You can literally interpret this model as a set of regression equations:

Y1 = b1*F + u1
Y2 = b2*F + u2
Y3 = b3*F + u3
Y4 = b4*F + u4