Step 1: Load breast cancer dataset from sklean and split it as train and test sets.

In [1]:
```
from sklearn.datasets import load_breast_cancer

dataset = load_breast_cancer()
print(dataset.feature_names)
print(dataset.target_names)
x = dataset.data
y = dataset.target
print(x.shape, y.shape)
```

```
['mean radius' 'mean texture' 'mean perimeter' 'mean area'
 'mean smoothness' 'mean compactness' 'mean concavity'
 'mean concave points' 'mean symmetry' 'mean fractal dimension'
 'radius error' 'texture error' 'perimeter error' 'area error'
 'smoothness error' 'compactness error' 'concavity error'
 'concave points error' 'symmetry error' 'fractal dimension error'
 'worst radius' 'worst texture' 'worst perimeter' 'worst area'
 'worst smoothness' 'worst compactness' 'worst concavity'
 'worst concave points' 'worst symmetry' 'worst fractal dimension']
['malignant' 'benign']
(569, 30) (569,)
```

In [2]:
```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25)
print(x_train.shape, x_test.shape)
print(y_train.shape, y_test.shape)
```

```
(426, 30) (143, 30)
(426,) (143,)
```

Step 2: Build the model

Import KNearestClassifier from sklearn.neighbors

--> Object creation: model = KNeighborsClassifier(n_neighbors)

In [3]:
```
from sklearn.neighbors import KNeighborsClassifier

model = KNeighborsClassifier(n_neighbors = 3)
model.fit(x_train, y_train)
```

Out[3]:
```
KNeighborsClassifier(n_neighbors=3)
```

Step 3: Test the model

We can import confusion_matrix, accuracy_score, recall_score, precision_score, f1_score functions from sklearn.metrics

In [4]:
```
from  sklearn.metrics import confusion_matrix, accuracy_score, recall_score, p

y_pred = model.predict(x_test)
print("Confusion matrix:\n", confusion_matrix(y_test, y_pred))
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Recall:", recall_score(y_test, y_pred))
print("Precision:", precision_score(y_test, y_pred))
print("F1 score:", f1_score(y_test, y_pred))
```

```
Confusion matrix:
 [[54  6]
 [ 4 79]]
Accuracy: 0.9300699300699301
Recall: 0.9518072289156626
Precision: 0.9294117647058824
F1 score: 0.9404761904761904
```