

In the model the building part, you can use the cancer dataset, which is a very famous multi-class classification dataset computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cells in the image.

The dataset comprises 30 features (mean radius, mean texture, mean perimeter, mean area, mean smoothness, mean concavity, mean concave points, mean symmetry, mean fractal dimension, radius error, texture error, perimeter error, compactness error, concavity error, concave points error, symmetry error, fractal dimension error, worst radius, worst perimeter, worst area, worst smoothness, worst compactness, worst concavity, worst concave points, worst symmetry, worst fractal dimension) and a target (type of cancer). This data has two types of cancer classes: malignant (harmful) and benign (not harmful). you can build a model to classify the type of cancer. The dataset is available in the scikit-learn library or you can download it from the UCI Machine Learning Library.

```
import numpy as np #linear algebra library of Python
from sklearn import datasets
```

```
cancer = datasets.load_breast_cancer()
```

```
cancer.data.shape
```



```
print(cancer.data[0:5])
```



```
from sklearn.model_selection import train_test_split #method to split training and test data
X_train, X_test, y_train, y_test=train_test_split(cancer.data, cancer.target, test_size=0.3)
```

```
from sklearn.ensemble import RandomForestClassifier
model=RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train,y_train)
y_pred=model.predict(X_test)
```

```
from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
from sklearn.metrics import confusion_matrix
y_pred=model.predict(X_test)
confusion_matrix(y_test,y_pred)
```



Excellent classifier: true positives=74, true negatives=38, false positives=2 and false negatives=0. Recall in our case is 100%. Precision is $TP/(TP+FP)=96.4\%$.