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

The dataset comprises 30 features (mean radius, mean texture, mean perimeter, mean area, mean smoothne
concavity, mean concave points, mean symmetry, mean fractal dimension, radius error, texture error, perimete
error, compactness error, concavity error, concave points error, symmetry error, fractal dimension error, worst
perimeter, worst area, worst smoothness, worst compactness, worst concavity, worst concave points, worst s
dimension) and a target (type of cancer). This data has two types of cancer classes: malignant (harmful) and
you can build a model to classify the type of cancer. The dataset is available in the scikit-learn library or you c
UCI Machine Learning Library.

```
import numpy as np #linear algebra library of Python
import pandas as pd # build on top of numpy for data analysis, data manipulation and da
import matplotlib.pyplot as plt #plotting library of Python
from sklearn import datasets
```

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

```
print(type(cancer))
```



```
cancer.data.shape
```



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



```
from sklearn.model_selection import train_test_split #method to split training and tes
X_train, X_test, y_train, y_test=train_test_split(cancer.data, cancer.target, test_siz
```

```
from sklearn.naive_bayes import MultinomialNB # smoothing is automatically applied
model=MultinomialNB()
model.fit(X_train, y_train)
score=model.score(X_test, y_test)
print("Accuracy:", score)
```



```
from sklearn import metrics
```

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

Classifier not so good: true positives=106, true negatives=50, false positives=13 and false negatives=2. Reca
98%. Precission is TP/(TP+FP)=89%