朴素贝叶斯模型

In [1]:

```
from pyspark.ml.classification import NaiveBayes
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.ml.feature import VectorAssembler
from pyspark.sql import SparkSession
import pandas as pd
```

In [2]:

```
# create a spark session
spark = SparkSession.builder.appName("NaiveBayes").getOrCreate()
```

In [3]:

```
# load the data
data = spark.read.format("csv").option("header", "true").option("inferSchema", "true").load("肿瘤

# create a vector assembler to combine the features into a single vector column
assembler = VectorAssembler(inputCols=data.columns[:-1], outputCol="features")

# transform the data using the vector assembler
data = assembler.transform(data)
data.show()
```

```
+--------+---------+---------+--------+--------+---------+--------+----------
---------+
|最大周长|最大凹陷度|平均凹陷度|最大面积|最大半径|平均灰度值|肿瘤性质|
features|
+--------+---------+---------+--------+--------+---------+--------+----------
---------+
|   184.6|   0.2654|   0.1471|  2019.0|   25.38|    17.33|       0|[184.6,0.26
54,0.1...|
|   158.8|    0.186|  0.07017|  1956.0|   24.99|    23.41|       0|[158.8,0.18
6,0.07...|
|   152.5|    0.243|   0.1279|  1709.0|   23.57|    25.53|       1|[152.5,0.24
3,0.12...|
|   98.87|   0.2575|   0.1052|   567.7|   14.91|     26.5|       0|[98.87,0.25
75,0.1...|
|   152.2|   0.1625|   0.1043|  1575.0|   22.54|    16.67|       0|[152.2,0.16
25,0.1...|
|   103.4|   0.1741|  0.08089|   741.6|   15.47|    23.75|       0|[103.4,0.17
41,0.0...|
|   153.2|   0.1932|    0.074|  1606.0|   22.88|    27.66|       0|[153.2,0.19
32,0.0...|
|   110.6|   0.1556|  0.05985|   897.0|   17.06|    28.14|       0|[110.6,0.15
56,0.0...|
|   106.2|    0.206|  0.09353|   739.3|   15.49|    30.73|       0|[106.2,0.20
6,0.09...|
|   97.65|    0.221|  0.08543|   711.4|   15.09|    40.68|       0|[97.65,0.22
1,0.08...|
|   123.8|  0.09975|  0.03323|  1150.0|   19.19|    33.88|       0|[123.8,0.09
975,0....|
|   136.5|    0.181|  0.06606|  1299.0|   20.42|    27.28|       0|[136.5,0.18
1,0.06...|
|   151.7|   0.1767|   0.1118|  1332.0|   20.96|    29.94|       0|[151.7,0.17
67,0.1...|
|   112.0|   0.1119|  0.05364|   876.5|   16.84|    27.66|       0|[112.0,0.11
19,0.0...|
|   108.8|   0.2208|  0.08025|   697.7|   15.03|    32.01|       0|[108.8,0.22
08,0.0...|
|   124.1|   0.1712|  0.07364|   943.2|   17.46|    37.13|       0|[124.1,0.17
12,0.0...|
|   123.4|   0.1609|  0.05259|  1138.0|   19.07|    30.88|       0|[123.4,0.16
09,0.0...|
|   136.8|   0.2073|   0.1028|  1315.0|   20.96|    31.48|       0|[136.8,0.20
73,0.1...|
|   186.8|   0.2388|  0.09498|  2398.0|   27.32|    30.88|       0|[186.8,0.23
88,0.0...|
|    99.7|   0.1288|  0.04781|   711.2|   15.11|    19.26|       1|[99.7,0.128
8,0.04...|
+--------+---------+---------+--------+--------+---------+--------+----------
---------+
only showing top 20 rows
```

In [4]:

```
# split the data into training and test sets
train, test = data.randomSplit([0.7, 0.3])
```

In [5]:

```
# create the classifier and fit it to the training data
nb = NaiveBayes(labelCol= '肿瘤性质',smoothing=1.0, modelType="multinomial")
model = nb.fit(train)
```

In [6]:

```
# make predictions on the test data
predictions = model.transform(test)

# evaluate the accuracy of the model
evaluator = MulticlassClassificationEvaluator(labelCol="肿瘤性质", predictionCol="prediction", m
accuracy = evaluator.evaluate(predictions)
print("Test set accuracy = " + str(accuracy))
```

```
Test set accuracy = 0.8830409356725146
```