Data Analytics

Yong Zheng

Illinois Institute of Technology
Chicago, IL, 60616, USA

School of Applied Technology
ILLINOIS INSTITUTE OF TECHNOLOGY

Review

- Supervised & Unsupervised Learning
- Classification
- KNN Classifier

Supervised v.s. Unsupervised Learning

- Supervised Learning: infer a (predictive) function from data associated with pre-defined targets/classes/labels
  Example: group objects by predefined labels
  Goal: Learn a model from labelled data (with multiple features) for future predictions
  Outcomes: We know outcomes: the predefined labels
  Evaluation: error/accuracy, and other more metrics
  Data Mining Task: Classification

- Unsupervised Learning: discover or describe underlying structure from unlabelled data
  Example: group objects by multiple features
  Goal: Learn the structure from unlabelled data (with multiple features)
  Outcomes: We do not know the outcomes
  Evaluation: No clear performance or evaluation methods
  Data Mining Task: Clustering

Supervised v.s. Unsupervised Learning

Machine Learning Algorithms *(sample)*

	Unsupervised	Supervised

Continuous	• Clustering & Dimensionality Reduction
	  ○ SVD
	  ○ PCA
	  ○ K-means
		• Regression
		  ○ Linear
		  ○ Polynomial
		• Decision Trees
		• Random Forests

Categorical	• Association Analysis
	  ○ Apriori
	  ○ FP-Growth
	• Hidden Markov Model
		• Classification
		  ○ KNN
		  ○ Trees
		  ○ Logistic Regression
		  ○ Naive-Bayes
		  ○ SVM

Supervised Learning: Classification

- Classification: a supervised way to group objects
  - We must have predefined labels
  - We must have knowledge: we know some instances are labeled by predefined classes/labels/categories
- For a Purpose of Prediction
  - To forecast or deduce the label/class based on values of features
  - Let the machines/computers think as humans
- There are many real-world applications
  - Financial Decision Making, e.g., credit card application
  - Image Processing, e.g., face recognition in cameras
  - Computer/Network Security, e.g., virus or attack detection
  - Information Retrieval, e.g., relevance of a document to a query
  - Recommender Systems, e.g., rating prediction for Amazon

Classification App: Credit Card Application

Terminologies in Classification

Features | classes

Age	Gender	Status	Income	Rent	Classes
27	Female	Student	$15,000	$800	Approved
32	Male	Part-time	$8,000	$400	Rejected
29	Male	Full-time	$50,000	$1200	?

Knowledge

Unseen data

Each row with features values is named as example or instance

Classification ➔ Learn from the knowledge (examples with unknown labels)
build predictive models to predict the unknown examples

Classification Task

There are usually three types of classification:

1). Binary Classification
Question: Is this an apple? Yes or No.

2). Multi-class Classification
Question: Is this an apple, banana or orange?

3). Multi-label Classification
Use appropriate words to describe it:
Red, Apple, Fruit, Tech, Mac, iPhone

KNN Classifier

- K-Nearest Neighbor (KNN) Classifier

A simple classifier, a lazy learner

1).Choose an odd number for K

2).Calculate distances between target and instances in training set

3).Pick the top KNN and assign the majority label as prediction

- Extended Problems in Classification Algorithms

Q1. Is it able to take categorical features? If Yes, how to treat them

Q2. Is normalization required?

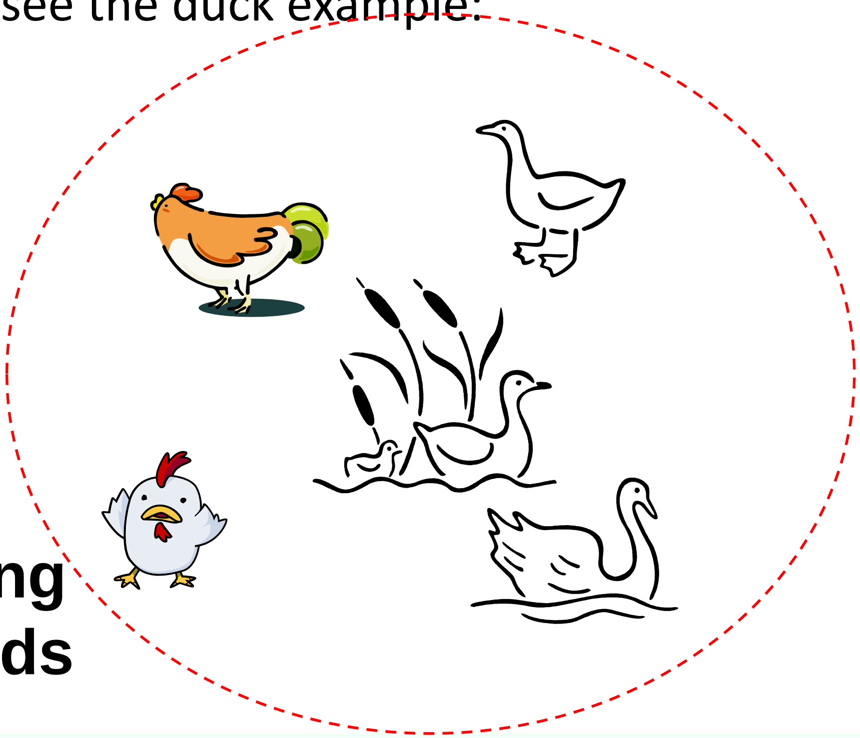
Q3. How to alleviate overfitting problem?

Note: they are general concerns in classification, not only KNN.

Naïve Bayes Classifier

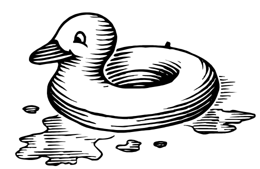
Naïve Bayes Classifier

- It is a probabilistic learning process
  - It is a simple classification algorithm too
  - You should have some preliminary knowledge about probability
  - There are some requirements to use the Naïve Bayes classifier
- Let's see the duck example:



Training Records

Unseen Data, E



Pr (duck | E) = ?
Pr (chicken | E) = ?

Basic Concepts In Probability I

- P(*A* | *B*) is the probability of *A* given *B;* *conditional probability*

Color	Weight (lbs)	Stripes	Tiger?
Orange	300	no	no
White	50	yes	no
Orange	490	yes	yes
White	510	yes	yes
Orange	490	no	no
White	450	no	no
Orange	40	no	no
Orange	200	yes	no
White	500	yes	yes
White	560	yes	yes

There are 10 examples here.
A: tiger = yes
B: color = orange

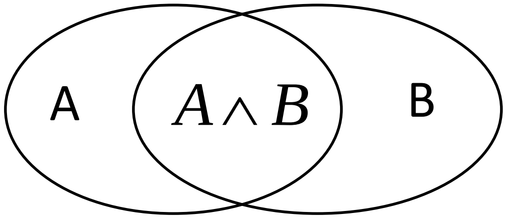
P (A) = 4/10 = 0.4
P (B) = 5/10 = 0.5
P (A | B) = ?

Color	Weight (lbs)	Stripes	Tiger?
Orange	300	no	no
Orange	490	yes	yes
Orange	490	no	no
Orange	40	no	no
Orange	200	yes	no

Basic Concepts In Probability II

- P(*A* | *B*) is the probability of *A* given *B*; *conditional probability*
- Assumes that *B* is all and only information known.
- Defined by:

$$\boxed{P(A \mid B) = \frac{P(A \wedge B)}{P(B)}}$$



- Bayes's Rule:

  Direct corollary of above definition

$$P(A \wedge B) = \frac{P(A \mid B)}{P(A)} = P(B \wedge A) = \frac{P(B \mid A)}{P(B)}$$

$$\Rightarrow P(A \mid B) = \frac{P(A)P(B \mid A)}{P(B)}$$

Naïve Bayes Classifier

- Let set of classes be $\{c_1, c_2, \ldots c_n\}$, e.g., c_1 = tiger, c_2 = lion
- Let E be description of an example (e.g., a vector with feature values)
- Determine class of E by computing for each class c_i

$$\boxed{P(c_i \mid E) = \frac{P(c_i)P(E \mid c_i)}{P(E)}}$$

- P(E) can be determined since classes are complete and disjoint:

$$\boxed{\sum_{i=1}^{n} P(c_i \mid E) = \sum_{i=1}^{n} \frac{P(c_i)P(E \mid c_i)}{P(E)} = 1}$$

$$\boxed{P(E) = \sum_{i=1}^{n} P(c_i)P(E \mid c_i)}$$

Naïve Bayes Classifier

- Determine class of *E* by computing for each class c_i

$$P(c_i \mid E) = \frac{P(c_i)P(E \mid c_i)}{P(E)}$$

$$P(E) = \sum_{i=1}^{n} P(c_i)P(E \mid c_i)$$

- Note: E is a feature vector, instead of a single feature!!

  For example:

  c: tiger = yes

$$E = e_1 \wedge e_2 \wedge \cdots \wedge e_m$$

  E: color = orange, weight = 500 lbs, stripes = yes

- Assume features are independent given the class (c_i), *conditionally independent;* Therefore, we then only need to know $P(e_j \mid c_i)$ for each feature and category [IMPORTANT Assumption!!!]

$$P(E \mid c_i) = P(e_1 \wedge e_2 \wedge \cdots \wedge e_m \mid c_i) = \prod_{j=1}^{m} P(e_j \mid c_i)$$

Conditional Independence

- X is conditionally independent of Y given Z, if the probability distribution for X is independent of the value of Y, given the value of Z
- Generally, $P(X,Y|Z) = P(X|Z) \times P(Y|Z)$

Let's say you flip two regular coins:
A - Your first coin flip is heads
B - Your second coin flip is heads
C - Your first two flips were the same

What is the relationship between A and B?
How about [A and B] by given C?

Conditional Independence

- X is conditionally independent of Y given Z, if the probability distribution for X is independent of the value of Y, given the value of Z
- Generally, $P(X,Y|Z) = P(X|Z) \times P(Y|Z)$

There are a regular coin and a fake one (two heads)
I randomly choose one of them and toss it twice
A - Your first flip is heads
B - Your second flip is heads
C - Your select a regular coin

What is the relationship between A and B?
How about [A and B] by given C?

Naïve Bayes Classifier

- Determine class of *E* by computing for each class c_i

$$P(c_i \mid E) = \frac{P(c_i)P(E \mid c_i)}{P(E)}$$

$$P(E) = \sum_{i=1}^{n} P(c_i)P(E \mid c_i)$$

- Note: E is a feature vector, instead of a single feature!!

  For example:

  c: tiger = yes

  $$E = e_1 \wedge e_2 \wedge \cdots \wedge e_m$$

  E: color = orange, weight = 500 lbs, stripes = yes

- Assume features are independent given the class (c_i), *conditionally independent;* Therefore, we then only need to know $P(e_j \mid c_i)$ for each feature and category [IMPORTANT Assumption!!!]

$$P(E \mid c_i) = P(e_1 \wedge e_2 \wedge \cdots \wedge e_m \mid c_i) = \prod_{j=1}^{m} P(e_j \mid c_i)$$

Example: Naïve Bayes Classifier

- c1: tiger = yes; c2: tiger = no

  E: $\underbrace{\text{color = orange}}_{e1}$, $\underbrace{\text{weight >= 500 lbs}}_{e2}$, $\underbrace{\text{stripes = yes}}_{e3}$

Color	Weight (lbs)	Stripes	Tiger?
Orange	500	no	no
White	50	yes	no
Orange	490	yes	yes
White	510	yes	yes
Orange	490	no	no
White	450	no	no
Orange	40	no	no
Orange	200	yes	no
White	500	yes	yes
White	560	yes	yes

$$\boxed{P(c1 \mid E) = \frac{P(c1)P(E \mid c1)}{P(E)}}$$

$$\boxed{P(E \mid c1) = \prod_{j=1}^{m} P(e_j \mid c1\ )}$$

P(e1 | c1) = ¼ = 0.25
P(e2 | c1) = ¾ = 0.75
P(e3 | c1) = 4/4 = 1
P(E | c1) = 0.25*0.75*1 = 0.1875

P(e1 | c2) = 4/6 = 0.667
P(e2 | c2) = 1/6 = 0.167
P(e3 | c2) = 2/6= 0.333
P(E | c2) = 0.0371

Example: Naïve Bayes Classifier

- c1: tiger = yes; c2: tiger = no

  E: color = orange, weight = 500 lbs, stripes = yes

  $\underbrace{\text{color = orange}}_{e1}$, $\underbrace{\text{weight = 500 lbs}}_{e2}$, $\underbrace{\text{stripes = yes}}_{e3}$

Color	Weight (lbs)	Stripes	Tiger?
Orange	500	no	no
White	50	yes	no
Orange	490	yes	yes
White	510	yes	yes
Orange	490	no	no
White	450	no	no
Orange	40	no	no
Orange	200	yes	no
White	500	yes	yes
White	560	yes	yes

$$P(c1 \mid E) = \frac{P(c1)P(E \mid c1)}{P(E)}$$

$$P(E \mid c1) = \prod_{j=1}^{m} P(e_j \mid c1)$$

P(E | c1) = 0.25*0.25*1 = 0.0625
P(E | c2) = 0.0371

$$P(E) = \sum_{i=1}^{n} P(c_i)P(E \mid c_i)$$

P(c1) = 4/10 = 0.4
P(c2) = 6/10 = 0.6
P(E) = P(c1)P(E|c1) + P(c2)P(E|c2) = 0.0473
P(c1|E) = 0.4*0.0625/0.0473 = 0.532

Example: Naïve Bayes Classifier

- c1: tiger = yes; c2: tiger = no

  $$\text{E: } \underbrace{\text{color = orange}}_{\text{e1}}, \underbrace{\text{weight = 500}}_{\text{e2}} \text{ lbs, } \underbrace{\text{stripes = yes}}_{\text{e3}}$$

Color	Weight (lbs)	Stripes	Tiger?
Orange	500	no	no
White	50	yes	no
Orange	490	yes	yes
White	510	yes	yes
Orange	490	no	no
White	450	no	no
Orange	40	no	no
Orange	200	yes	no
White	500	yes	yes
White	560	yes	yes

P(E | c1) = 0.25*0.25*1 = 0.0625
P(E | c2) = 0.0371
P(c1) = 4/10 = 0.4
P(c2) = 6/10 = 0.6
P(E) = P(c1)P(E|c1) + P(c2)P(E|c2) = 0.0473
P(c1|E) = 0.4*0.0625/0.0473 = 0.532

$$P(c2 \mid E) = \frac{P(c2)P(E \mid c2)}{P(E)}$$

P(c2|E) = 0.6*0.0371/0.0473 = 0.471

Example: Naïve Bayes Classifier

- c1: tiger = yes; c2: tiger = no

  E: $\underbrace{\text{color = orange}}_{e1}$, $\underbrace{\text{weight = 500 lbs}}_{e2}$, $\underbrace{\text{stripes = yes}}_{e3}$

Color	Weight (lbs)	Stripes	Tiger?
Orange	500	no	no
White	50	yes	no
Orange	490	yes	yes
White	510	yes	yes
Orange	490	no	no
White	450	no	no
Orange	40	no	no
Orange	200	yes	no
White	500	yes	yes
White	560	yes	yes

P(c1|E) = 0.4*0.0625/0.0473 = 0.532
P(c2|E) = 0.6*0.0371/0.0473 = 0.471

P(c1|E) > P(c2|E)

We have more confidence to say we should trust c1

In other words, E should be classified as tiger!!!!

Naïve Bayes Classifier

It is very useful. A list of applications:

- Medical Detection: Given the situation of patients (such as cough, headache, body temp, etc), make a decision he or she is in disease or not.
- Gender Classification: Given weights, age, heights, size of feet to judge a person is male or female
- Social Robots: Given use behaviors on social networks, such as how many posts, how many friends, whether they use real human icons, the daily frequency of posts or friends, to make a decision this account is a real one or a fake one
- Text Classification: such as news or email classification

Naïve Bayes Classifier

Solutions to Improve Naïve Bayes Classifier

- Imbalance Issue: In the training set, knowledge are imbalanced
- Numerical Features

Color	Weight (lbs)	Stripes	Tiger?
Orange	500	no	no
White	50	yes	no
Orange	490	yes	yes
White	510	yes	yes
Orange	490	no	no
White	450	no	no
Orange	40	no	no
Orange	200	yes	no
White	500	yes	yes
White	560	yes	yes

Weights = 500
Weights > 500

Which one is better?

KNN vs Naïve Bayes Classifier

1). How to treat numerical and categorical data in Naïve Bayes?

In KNN, we need to transform nominal data to numerical ones.
In Naïve Bayes, we need to transform numerical data to nominal data.

2). Is normalization required in Naïve Bayes?

It is not necessary in Naïve Bayes, but required in KNN

3). Overfitting in Naïve Bayes?

Be careful about the imbalanced data in Naïve Bayes.

4). Which one is better?

It depends. You'd better try Naïve Bayes if there are multiple nominal features

In-Class Practice

Outlook	Tempreature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

- Here, we have two classes C1="yes" (Positive) and C2="no" (Negative)
- Suppose we have new instance X = <sunny, mild, high, true>. How should it be classified?
- Compare Pr(P|X) and Pr(N|X)

Classification By Naïve Bayes

- Iris Data

Download if from UCI ML Repository

https://archive.ics.uci.edu/ml/datasets/iris

- Features and label

You need to read the page to understand the features and labels

Label = Setosa, Versicolour, Virginica

Features:

Sepal length and width, Petal length and width

You need to explore the feature data types one by one

In this data, all of the features are numerical variables

- Libraries for naïve bayes

There are many libraries you can use, e.g, naivebayes and caret

Classification By Naïve Bayes

- Load data and library

```
install.packages('naivebayes', dependencies = TRUE)
library(naivebayes)
data(iris)
head(iris)
names(iris)
ind_iris <- sample(1:nrow(iris), size = round(0.3 * nrow(iris)))
iris_train <- iris[-ind_iris, ]
iris_test <- iris[ind_iris, ]
```

30% as testing

+/- Output

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Classification By Naïve Bayes

- Build Models and make predictions

nb_iris <- naive_bayes(Species ~ ., iris_train)

pred=predict(nb_iris, iris_test)

head(predict(nb_iris, iris_test, type = "prob"))

```
> nb_iris <- naive_bayes(Species ~ ., iris_train)
> predict(nb_iris, iris_test)
 [1] virginica  setosa     setosa     setosa     setosa
[16] versicolor virginica  virginica  versicolor virginica
[31] virginica  virginica  virginica  virginica  versicolor
Levels: setosa versicolor virginica
> head(predict(nb_iris, iris_test, type = "prob"))
            setosa   versicolor    virginica
[1,] 9.437529e-161 7.912698e-06 9.999921e-01
[2,]  1.000000e+00 3.544556e-20 2.519968e-24
[3,]  1.000000e+00 4.684493e-19 2.091121e-23
[4,]  1.000000e+00 5.939605e-19 3.047624e-23
[5,]  1.000000e+00 6.663923e-18 1.082599e-21
[6,]  1.000000e+00 7.431968e-19 3.004861e-23
```

Classification By Naïve Bayes

- Evaluate performance

install.packages('Metrics', dependencies = TRUE)

library(Metrics)

accuracy(iris_test[,5], pred)

Actual data

predictions

```
> library(Metrics)

Attaching package: 'Metrics'

The following objects are masked from 'pa

    precision, recall

> accuracy(iris_test[,5], pred)
[1] 1
```

Classification By Naïve Bayes

- Load data and library for N-fold cross validation

```
install.packages('caret', dependencies = TRUE)
library(caret)
head(iris)
names(iris)
```

+/- Output

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Classification By Naïve Bayes

- Set features and labels

```
x = iris[,-5]
y = iris$Species
```

+/- Output

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Classification By Naïve Bayes

- Build the model with 10-Folds cross validation

model = train(x,y,'nb',trControl=trainControl(method='cv',number=10),na.action=na.pass)

The train function is very powerful
You can use several classification methods and evaluation methods

For more details
https://machinelearningmastery.com/how-to-estimate-model-accuracy-in-r-using-the-caret-package/

Classification By Naïve Bayes

- Make the predictions

predict(model$finalModel,x)

+/- Output

```
## $class
##   [1] setosa     setosa     setosa     setosa     setosa     setosa
##   [7] setosa     setosa     setosa     setosa     setosa     setosa
##  [13] setosa     setosa     setosa     setosa     setosa     setosa
##  [19] setosa     setosa     setosa     setosa     setosa     setosa
##  [25] setosa     setosa     setosa     setosa     setosa     setosa
##  [31] setosa     setosa     setosa     setosa     setosa     setosa
##  [37] setosa     setosa     setosa     setosa     setosa     setosa
##  [43] setosa     setosa     setosa     setosa     setosa     setosa
##  [49] setosa     setosa     versicolor versicolor versicolor versicolor
##  [55] versicolor versicolor versicolor versicolor versicolor versicolor
##  [61] versicolor versicolor versicolor versicolor versicolor versicolor
##  [67] versicolor versicolor versicolor versicolor virginica  versicolor
##  [73] versicolor versicolor versicolor versicolor versicolor virginica
##  [79] versicolor versicolor versicolor versicolor versicolor virginica
##  [85] versicolor versicolor versicolor versicolor versicolor versicolor
```

Classification By Naïve Bayes

- Make the predictions

predict(model$finalModel,x)

```
## $posterior
##            setosa versicolor virginica
##   [1,] 1.000e+00  3.122e-09 8.989e-11
##   [2,] 1.000e+00  4.953e-08 1.362e-09
##   [3,] 1.000e+00  1.950e-08 1.153e-09
##   [4,] 1.000e+00  1.146e-08 6.617e-10
##   [5,] 1.000e+00  8.840e-10 8.567e-11
##   [6,] 1.000e+00  3.819e-09 5.966e-09
##   [7,] 1.000e+00  7.394e-09 6.703e-10
##   [8,] 1.000e+00  5.312e-09 1.920e-10
##   [9,] 1.000e+00  6.502e-09 3.194e-10
##  [10,] 1.000e+00  1.732e-07 5.532e-09
##  [11,] 1.000e+00  1.234e-09 4.373e-10
##  [12,] 1.000e+00  6.937e-09 4.553e-10
##  [13,] 1.000e+00  2.398e-07 8.627e-09
##  [14,] 1.000e+00  1.001e-07 5.967e-09
##  [15,] 1.000e+00  1.005e-08 1.607e-08
##  [16,] 1.000e+00  2.410e-08 1.560e-09
##  [17,] 1.000e+00  2.068e-09 3.230e-09
```

Classification By Naïve Bayes

- Get accuracy

print(model)

```
> print(model)
Naive Bayes

150 samples
  4 predictor
  3 classes: 'setosa', 'versicolor', 'virginica'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 135, 135, 135, 135, 135, 135, ...
Resampling results across tuning parameters:

  usekernel  Accuracy   Kappa
  FALSE      0.9533333  0.93
   TRUE      0.9600000  0.94

Tuning parameter 'fL' was held constant at a value of 0
Tuning parameter 'adjust' was held constant at a value of 1
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were fL = 0, usekernel = TRUE and adjust = 1.
```

Classification By Naïve Bayes

- Preprocessing to improve your models

For any numerical features, it is better to convert them to nominal

F1	F2	F3	F4	Class
C3	0	0	2	—
C2	1	0	5	+
C1	0	1	8	—
C2	1	1	16	—
C1	1	0	23	+
C3	0	1	11	+

Usually we use the cut function to create N groups

Data = cut(dataColumn, N)

```
> data=read.table("book1.csv", head=T, sep=',')
> data[,4]
[1]  2  5  8 16 23 11
> data[,4]=cut(data[,4], 3)
> data[,4]
[1] (1.98,9] (1.98,9] (1.98,9] (9,16]   (16,23]  (9,16]
Levels: (1.98,9] (9,16] (16,23]
> head(data)
  F1 F2 F3       F4 Class
1 C3  0  0 (1.98,9]     —
2 C2  1  0 (1.98,9]     +
3 C1  0  1 (1.98,9]     —
4 C2  1  1   (9,16]     —
5 C1  1  0  (16,23]     +
6 C3  0  1   (9,16]     +
```