Week – 5
Naïve Bayes

Characteristics

Data-driven, not model-driven

Make no assumptions about the data

Naïve Bayes: The Basic Idea

For a given new record to be classified, find other records like it (i.e., same values for the predictors)

What is the prevalent class among those records?

Assign that class to your new record

Exact Bayes Classifier

Relies on finding other records that share <u>same predictor values</u> as record-to-be-classified.

Want to find "probability of belonging to class *C*, given specified values of predictors."

Even with large data sets, may be hard to find other records that **exactly match** your record, in terms of predictor values.

Solution – Naïve Bayes

- Assume independence of predictor variables (within each class)
- Use multiplication rule
- Find same probability that record belongs to class C, given predictor values

Bayes rule

$$\boldsymbol{P(c|x) = \frac{P(x|c) * P(c)}{P(x)}}$$

$\boldsymbol{P(c|x)}$ – is the posterior probability of *class* (*target*) given *predictor* (*attribute*).

$\boldsymbol{P(x|c)}$ – is the likelihood which is the probability of *predictor* given *class*

$\boldsymbol{P(c)}$ – is the prior probability of *class*.

$\boldsymbol{P(x)}$ – is the prior probability of *predictor*

Bayes rule: example

$$P(c|x) = \frac{P(x|c) * P(c)}{P(x)}$$

Given

- A doctor knows that meningitis causes stiff neck 50% of the time
- Prior probability of any patient having meningitis is 1/50,000
- Prior probability of any patient having stiff neck is 1/20

If a patient has stiff neck,
what's the probability he/she has meningitis ?

Bayesian classifier

Classify **c to the class which has bigger posterior probability**

$$c^* = h_c(x) = arg \max_{j=1.....m} P(c_j|x)$$

We will use Bayes theorem to determine $P(c_j|x)$

$$P(c|x) = \frac{P(x|c) * P(c)}{P(x)}$$

$\boldsymbol{P(x)}$ is the same for all categories so can be ignored

$$c^* = h_c(x) = arg \max_{j=1.....m} P\left(c_j\right) * P(x|c_j)$$

Naïve Bayes classifier

"Naïve" because of its very naïve independence assumption:

all the attributes are conditionally independent given the class

$$P(x|c_j) = \prod_{i=1}^{n} P(X_i = x_k|c_j)$$

Because of the assumption of independence

$$\boldsymbol{i - number\ of\ variables,}$$
$$\boldsymbol{j - number\ of\ classes}$$

Classification rule

$$c^* = h_c(x) = arg \max_{j=1.....m} P(c_j) \prod_{i=1}^{n} P(X_i = x_i|c_j)$$

Naïve Bayes classifier: Algorithm

1. Estimate probability $P(c_j)$ for each class j.

$P(c_j) = \frac{N_j}{N}$, where N_j is the number of cases from class j

2. If the independent variable is **categorical**, then calculate the conditional probability

$P(X_i = x_k | c_j) = \frac{N_{ijk}}{N_j}$, where N_{ijk} number of examples of the class c_j having value x_k for the variable X_i

3. If the independent variable is continuous there are two options:

- Make it categorical and do calculations as in step 2
- Assume normal distribution per class

$P\left(X_i = x_k \middle| c_j\right) = N(X_i = x_k|\mu_{ij}, \sigma_{ij})$, where μ_{ij}, σ_{ij} are the means and standard deviations of the variable i for each class j.

Naïve Bayes: example

Having the following contingency tables, build Naïve Bayes model to predict the class label and respective probabilities of survival for the following person:

{Sex=Female, Class=1}

	No	Yes	Sum
female	127	339	466
male	682	161	843
Sum	809	500	1309

	No	Yes	Sum
1	123	200	323
2	158	119	277
3	528	181	709
Sum	809	500	1309

Naïve Bayes: example

$Calculate\ the\ value\ of\ h_c(x)$

$$h_{Yes}(Female, Pclass1) = P(sex = Female|Yes) * P(Pclass = 1|Yes) * P(Yes)$$

$h_{No}(Female, Pclass1)$ -??

Naïve Bayes: example

$$Calculate\ the\ value\ of\ h_c(x)$$

$$h_{Yes}(Female, Pclass1) = P(sex = Female|Yes) * P(Pclass = 1|Yes) * P(Yes)$$

$$h_{No}(Female, Pclass1) = P(sex = Female|No) * P(Pclass = 1|No) * P(No)$$

Naïve Bayes: example

$$h_{Yes}(Female, Pclass1) = P(sex = Female|Yes) * P(Pclass = 1|Yes) * P(Yes)$$

	No	Yes	Sum
female	127	339	466
male	682	161	843
Sum	809	500	1309

	No	Yes	Sum
1	123	200	323
2	158	119	277
3	528	181	709
Sum	809	500	1309

Naïve Bayes: example

$$h_{Yes}(Female, Pclass1) = P(sex = Female|Yes) * P(Pclass = 1|Yes) * P(Yes)$$

	No	Yes	Sum
female	127	339	466
male	682	161	843
Sum	809	500	1309

	No	Yes	Sum
1	123	200	323
2	158	119	277
3	528	181	709
Sum	809	500	1309

$$P(sex = Female|Yes) = \frac{339}{500} = 0.678$$

$$P(Pclass = 1|Yes) = \frac{200}{500} = 0.4$$

$$P(Yes) = \frac{500}{1309} = 0.38$$

$$h_{Yes}(Female, Pclass1) = 0.678 * 0.4 * 0.38 = 0.1$$

Naïve Bayes: example

$$h_{No}(Female, Pclass1) = P(sex = Female|No) * P(Pclass = 1|No) * P(No)$$

	No	Yes	Sum
female	127	339	466
male	682	161	843
Sum	809	500	1309

	No	Yes	Sum
1	123	200	323
2	158	119	277
3	528	181	709
Sum	809	500	1309

Naïve Bayes: example

$$h_{No}(Female, Pclass1) = P(sex = Female|No) * P(Pclass = 1|No) * P(No)$$

	No	Yes	Sum
female	127	339	466
male	682	161	843
Sum	809	500	1309

	No	Yes	Sum
1	123	200	323
2	158	119	277
3	528	181	709
Sum	809	500	1309

$$P(sex = Female|No) = \frac{127}{809} = 0.157$$

$$P(Pclass = 1|No) \ = \frac{123}{809} = 0.152$$

$$P(No) = \frac{809}{1309} = 0.61$$

$$h_{No}(Female, Pclass1)$$
$$= 0.157 * 0.152 * 0.61 = 0.1 = 0.014$$

Naïve Bayes: example

$$h_{Yes}(x) = 0.1$$

$$h_{No}(x) = 0.014$$

$\max\left(h_{Yes}(x), h_{No}(x)\right)$ is for yes so we will predict class yes

Naïve Bayes: example

What about probability ?

Using bayes theorem and the **regularization** parameter

$$P(survived = Yes|Female, Pclas1) = \frac{h_{Yes}(x)}{h_{Yes}(x) + h_{No}(x)} = \frac{0.1}{0.1 + 0.014} = 0.87$$

$$P(survived = No|Female, Pclas1) = \frac{h_{No}(x)}{h_{Yes}(x) + h_{No}(x)} = \frac{0.002}{0.1 + 0.014} = 0.13$$

Example

RID	age	income	student	credit	C_i: buy
1	youth	high	no	fair	C_2: no
2	youth	high	no	excellent	C_2: no
3	middle-aged	high	no	fair	C_1: yes
4	senior	medium	no	fair	C_1: yes
5	senior	low	yes	fair	C_1: yes
6	senior	low	yes	excellent	C_2: no
7	middle-aged	low	yes	excellent	C_1: yes
8	youth	medium	no	fair	C_2: no
9	youth	low	yes	fair	C_1: yes
10	senior	medium	yes	fair	C_1: yes
11	youth	medium	yes	excellent	C_1: yes
12	middle-aged	medium	no	excellent	C_1: yes
13	middle-aged	high	yes	fair	C_1: yes
14	senior	medium	no	excellent	C_2: no

The data samples are described by attributes age, income, student, and credit. The class label attribute, buy, tells whether the person buys a computer, has two distinct values, yes (class C_1) and No (class C_2).

Example

RID	age	income	student	credit	C_i: buy
1	youth	high	no	fair	C_2: no
2	youth	high	no	excellent	C_2: no
3	middle-aged	high	no	fair	C_1: yes
4	senior	medium	no	fair	C_1: yes
5	senior	low	yes	fair	C_1: yes
6	senior	low	yes	excellent	C_2: no
7	middle-aged	low	yes	excellent	C_1: yes
8	youth	medium	no	fair	C_2: no
9	youth	low	yes	fair	C_1: yes
10	senior	medium	yes	fair	C_1: yes
11	youth	medium	yes	excellent	C_1: yes
12	middle-aged	medium	no	excellent	C_1: yes
13	middle-aged	high	yes	fair	C_1: yes
14	senior	medium	no	excellent	C_2: no

Classify the following object

X = (age = youth, income = medium, student = yes, credit = fair)

Laplacian correction

What if for one class the attribute for an independent variable is missing in the training dataset?

RID	age	income	student	credit	C_i: buy
1	youth	high	no	fair	C_2: no
2	youth	high	no	excellent	C_2: no
3	middle-aged	high	no	fair	C_1: yes
4	senior	medium	no	fair	C_1: yes
5	senior	low	yes	fair	C_1: yes
6	senior	low	yes	excellent	C_2: no
7	middle-aged	low	yes	excellent	C_1: yes
8	youth	medium	no	fair	C_2: no
9	youth	low	yes	fair	C_1: yes
10	senior	medium	yes	fair	C_1: yes
11	youth	medium	yes	excellent	C_1: yes
12	middle-aged	medium	no	excellent	C_1: yes
13	middle-aged	high	yes	fair	C_1: yes
14	senior	medium	no	excellent	C_2: no

Laplacian correction

What if for one class the attribute for an independent variable is missing in the training dataset?

RID	age	income	student	credit	C_i: buy
1	youth	high	no	fair	C_2: no
2	youth	high	no	excellent	C_2: no
3	middle-aged	high	no	fair	C_1: yes
4	senior	medium	no	fair	C_1: yes
5	senior	low	yes	fair	C_1: yes
6	senior	medium	yes	excellent	C_2: no
7	middle-aged	low	yes	excellent	C_1: yes
8	youth	medium	no	fair	C_2: no
9	youth	low	yes	fair	C_1: yes
10	senior	medium	yes	fair	C_1: yes
11	youth	medium	yes	excellent	C_1: yes
12	middle-aged	medium	no	excellent	C_1: yes
13	middle-aged	high	yes	fair	C_1: yes
14	senior	medium	no	excellent	C_2: no

Laplacian correction: Solution

Solution:
Add constant number to zero frequency categories (don't forget to increase total sample size by the same number).

Results:
No categories with zero frequency

Doing in R

Doing in R

Call the libraries and read the csv file

```
library(caret)
library(e1071)
library(ROCR)
hr_data<-read.csv("HR_balanced.csv")
```

Create testing and training sets

```
set.seed(1)
index<-createDataPartition(hr_data$left, p=0.75, list=F)
Train<-hr_data[index,]
Test<-hr_data[-index,]
```

The variable left is our dependent variable (values: ‘No’, ‘Yes’)

```
library(e1071)
Model<-naiveBayes(left~., data=Train,
                  laplace=1)
```

```
names(Model)
```

```
## [1] "apriori" "tables"  "levels"  "call"
```

The resulting object Model is a list look what is inside

Doing in R

table for apriori probabilties

```
Model$apriori
```

```
## Y
##   No  Yes
## 3000 2679
```

$$P(No) = \frac{3000}{3000 + 2679} = 0.52$$

$$P(Yes) = \frac{2679}{3000 + 2679} = 0.48$$

Doing in R

Model$tables gives the conditional probabilities

```
## $Work_accident
##      Work_accident
## Y        Accident No accident
##   No  0.16922052  0.83077948
##   Yes 0.04774338  0.95225662
##
## $promotion_last_5years
##      promotion_last_5years
## Y       No promotion   Promotion
##   No    0.972351765 0.027648235
##   Yes   0.993659082 0.006340918
##
```

$$\boxed{P(Accident|No) = 0.16}$$

$$\boxed{P(No\ Accident|Yes) = 0.95}$$

Doing in R

Model$table

For numeric variables gives the mean and standard deviations that can be used to predict probabilities using Normal Gaussian distribution

```
Model$tables
```

```
## $satisfaction_level
##       satisfaction_level
## Y             [,1]      [,2]
##   No  0.6654067 0.2151299
##   Yes 0.4442777 0.2651830
##
## $last_evaluation
##       last_evaluation
## Y             [,1]      [,2]
##   No  0.7185933 0.1639149
##   Yes 0.7198096 0.1978558
```

Mean

Standard Deviation

Doing in R

You can also access the individual tables

```
Model$tables$salary
```

```
##      salary
## Y          high       low    medium
##   No  0.1052281 0.4402264 0.4545455
##   Yes 0.0246085 0.6058911 0.3695004
```

```
Model$tables$average_montly_hours
```

```
##      average_montly_hours
## Y         [,1]     [,2]
##   No  199.1630 45.42868
##   Yes 207.9168 61.09058
```

Doing in R

```
pred_test<-predict(Model, newdata=Test)
confusionMatrix(pred_test, Test$left, positive="Yes")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  No Yes
##        No  831 225
##        Yes 168 667
##
##                Accuracy : 0.7922
##                  95% CI : (0.7732, 0.8103)
##     No Information Rate : 0.5283
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5816
##  Mcnemar's Test P-Value : 0.004731
##
##             Sensitivity : 0.7478
##             Specificity : 0.8318
##          Pos Pred Value : 0.7988
##          Neg Pred Value : 0.7869
##              Prevalence : 0.4717
##          Detection Rate : 0.3527
##    Detection Prevalence : 0.4416
##       Balanced Accuracy : 0.7898
##
##        'Positive' Class : Yes
```

Prediction and the confusion matrix

Doing in R

If you want to get the probabilities predicted then use type='raw' in the predict command: The result is a matrix with two columns

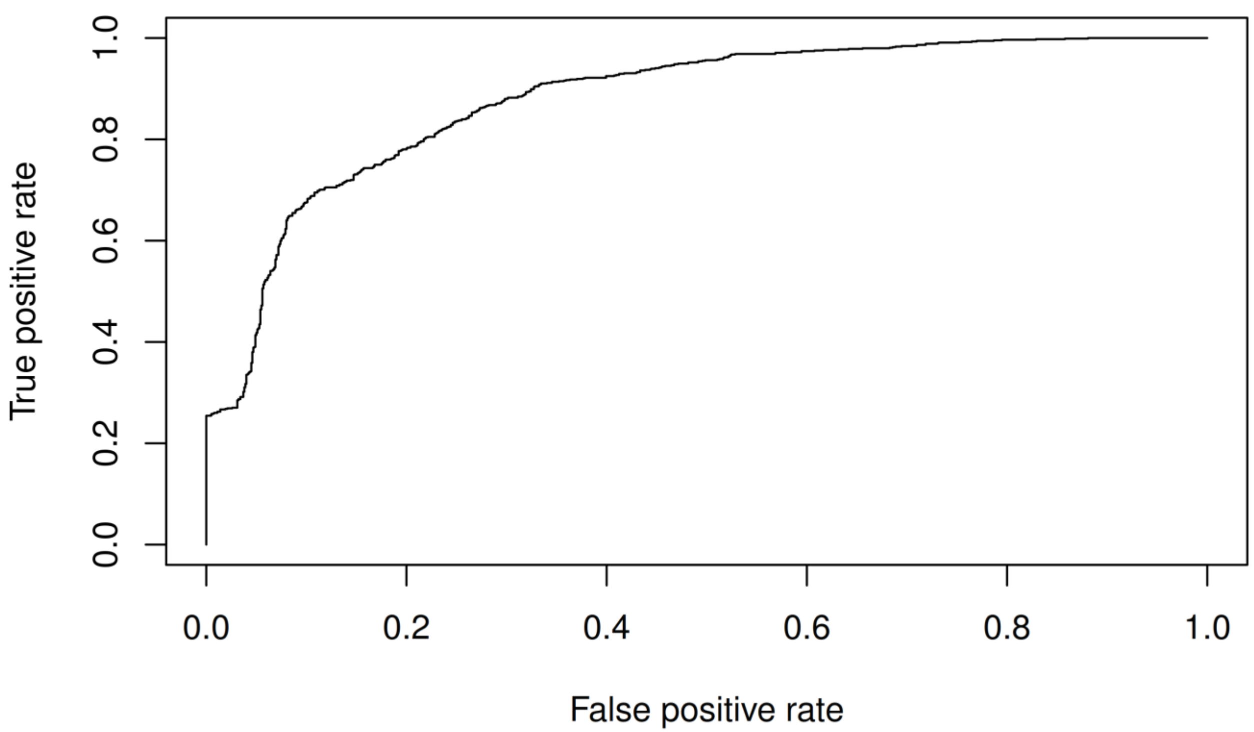
```
pred_test_prob<-predict(Model, newdata=Test,
                        type="raw")
head(pred_test_prob)
```

```
##               No       Yes
## [1,] 0.789718255 0.2102817
## [2,] 0.180254942 0.8197451
## [3,] 0.012005639 0.9879944
## [4,] 0.216939908 0.7830601
## [5,] 0.175418814 0.8245812
## [6,] 0.006869546 0.9931305
```

Doing in R

For the ROCR curve prediction command use probabilities for the positive case ('Yes')

```
p_test<-prediction(pred_test_prob[,2], Test$left)
perf<-performance(p_test, "tpr","fpr")
plot(perf)
```



Doing in R

Area under the curve

```
performance(p_test, "auc")@y.values
```

```
## [[1]]
## [1] 0.8783088
```