MACHINE LEARNING

ASSIGNMENT – 2

NAME: SARITHA H

ROLL NO: 25

Naive Bayes Classifiers

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Example

In our example case, we will work on a data having 9134 records of customers. The attributes about each customers provided are "Customer ID", "State", "Education", "Employment Status", "Gender", "Location", "Marital Status", "Vehicle" and "Income" .The objective of building our model is to predict the income level of customers. The income is divided into two levels, high and low. The assumption being that the customers having income below 35000 is considered as the low-income customer, and those having income more than 35000 are high-income customers.

The steps to be followed for the model building :

1. Import the data.

2. Data cleaning is an important part.

3. Creating a derived column with respect to the income column. The new column indicates only the income levels (high or low), based on the assumption made above.

4. Divide the data in 7:3 ratio. First part is training data that will be used to make the machine learn the data trend. The second part is to predict their income levels.

5. Then comes the step to see the predictions made by the model and check how accurate these predictions are.

So as explained above we start with our model building from the first step onwards.

Import Data

```
> data = read.csv("C:\\Program Files\\R\\R-3.6.2\\input.csv")
```

Data Cleaning

As mentioned above, our target is to predict the income levels of customers. So we create a column stating the income levels, i.e., high and low, according to the income mentioned.Let us

set that if a customer has income more than 35000, then we keep him in the “high” slot, otherwise we set him “low”.

```
> data1 <- data

> data1$inc <- ifelse(data$Income >= 35000, "High", "Low")
```

Now, we remove the 9th variable (Income) as we are itself taking the income levels as a calculated field. Few variables that are irrelevant with regards to this model should be removed. These may include, “Customer”, “Gender” and “Marital Status”. These variables should have no direct connections on determining the income of the customers.

```
> data1 <- data1[, -9]

> data1 <- data1[, c(-1,-5,-7)]
```

Checking the structure of the variables,

```
> str(data1)

'data.frame':   9134 obs. of  6 variables:

 $ State          : Factor w/ 5 levels "Arizona","California",..: 5 1 3 2 5 4 4 1 4 4 ...

 $ Education      : Factor w/ 5 levels "Bachelor","College",..: 1 1 1 1 1 1 2 5 1 2 ...

 $ EmploymentStatus: Factor w/ 5 levels "Disabled","Employed",..: 2 5 2 5 2 2 2 5 3 2 ...

 $ Location       : Factor w/ 3 levels "Rural","Suburban",..: 2 2 2 2 1 1 2 3 2 3 ...

 $ Vehicle        : Factor w/ 6 levels "Four-Door Car",..: 6 1 6 5 1 6 1 1 1 1 ...

 $ inc            : chr  "High" "Low" "High" "Low" ...
```

We see there are 9134 records and 6 variables structured in a data frame. Only the problem is that the variable “inc” in the char data type, which is a problem. As there are only two levels in this variable, high and low, hence we have to convert it into factor data type.

```
> data1$inc <- as.factor(data1$inc)
```

Naive Bayes Classifier Model

```
> library(e1071)

Warning message:
```

package ‘e1071’ was built under R version 3.6.3

> library(caret)

Loading required package: lattice

Loading required package: ggplot2

Warning messages:

1: package ‘caret’ was built under R version 3.6.3

2: package ‘ggplot2’ was built under R version 3.6.3

> 1

[1] 1

> 2

[1] 2

> 3

[1] 3

> data1$inc <- as.factor(data1$inc)

> set.seed(2)

> random <- sample(2, nrow(data1), prob = c(0.7, 0.3), replace = T)

> data_train <- data1[random == 1,]

> data_test <- data1[random == 2,]

Running the naive Bayes function. Keeping “inc” as the dependent variable and considering all other 5 variables as independent variables (indicated with “.” sign). Running the model on the training set first.

> data_nb <- naiveBayes(inc ~ . , data = data_train)

On running “data_nb” we get to see the summary of the model run. We read it as,

Under the heading “A-priori probabilities”, we see that there is 49% chance of income of the testing dataset customers being low. Similarly 51% chance of income of the testing dataset customers being high.Under the heading “Conditional probabilities”, we get the conditional probabilities of all the variables individually.If the State is “Arizona”, the probability of the income being high is more than the probability of the income being low. Similarly, if the state is “California”, the probability of the income being low is more than the probability of the income is high. We read the rest in this manner.Next, if the Education is “Bachelor”, the probability of the income being low is more than the probability of the income is high. Compared to “Master”, the probability of the income being high is much more than the probability of the income is low, which is logical.We can read the other observations in the same way.Now running the model on the test data and getting the predictions,

```
> data_nb

Naive Bayes Classifier for Discrete Predictors

Call:

naiveBayes.default(x = X, y = Y, laplace = laplace)

A-priori probabilities:

Y

    High      Low

0.4865418 0.5134582

Conditional probabilities:

     State

Y       Arizona California     Nevada     Oregon Washington

 High 0.19184730 0.33128437 0.09899709 0.28566807 0.09220317

 Low  0.18485592 0.35591662 0.09564684 0.27774372 0.08583691

 Education

Y       Bachelor    College     Doctor High School or Below     Master

 High 0.28987383 0.29537367 0.04496927          0.27499191 0.09479133

 Low  0.29736358 0.29521766 0.03402820          0.31054568 0.06284488
```

EmploymentStatus

Y	Disabled	Employed	Medical Leave	Retired	Unemployed
High	0.00000000	1.00000000	0.00000000	0.00000000	0.00000000
Low	0.08645003	0.26762722	0.09442060	0.06069896	0.49080319

Location

Y	Rural	Suburban	Urban
High	0.32125526	0.40601747	0.27272727
Low	0.07970570	0.84334764	0.07694666

Vehicle

Y	Four-Door Car	Luxury Car	Luxury SUV	Sports Car	SUV Two-Door Car	
High	0.51277904	0.02167583	0.01876415	0.05370430	0.18117114	0.21190553
Low	0.50367872	0.01686082	0.02237891	0.06008584	0.20202330	0.19497241

Now running the model on the test data and getting the predictions,The variable “pred_nb” stores the high and low levels corresponding to all the records. To read it properly let’s create a confusion matrix out of it,

```
> pred_nb <- predict(data_nb, data_test)
> confusionMatrix(table(pred_nb, data_test$inc))
Confusion Matrix and Statistics
pred_nb High  Low
  High 1382  352
  Low     0 1047
    Accuracy : 0.8734
     95% CI : (0.8605, 0.8856)
   No Information Rate : 0.5031
   P-Value [Acc > NIR] : < 2.2e-16
```

Kappa : 0.7472

Mcnemar's Test P-Value : < 2.2e-16

Sensitivity : 1.0000

Specificity : 0.7484

Pos Pred Value : 0.7970

Neg Pred Value : 1.0000

Prevalence : 0.4969

Detection Rate : 0.4969

Detection Prevalence : 0.6235

Balanced Accuracy : 0.8742

'Positive' Class : High