Universidad del Valle de Guatemala
Minería de Datos
Lynette García
17/03/2019

Andrea María Cordón Mayén, 16076
Cristopher Sebastian Recinos Ramírez, 16005
Fecha de Entrega:

Hoja de Trabajo 5
"Naive Bayes"

1. **Elaborar un modelo de bayes ingenuo utilizando el conjunto de entrenamiento y explicar los resultados obtenidos.**
    a. El modelo de naive bayes se elaboró utilizando las variables AdoptionSpeed y Type. Como se puede observar, el modelo realizado quedó relativamente bien ya los datos no están distribuidos uniformemente que. El modelo se elaboró utilizando el siguiente código:

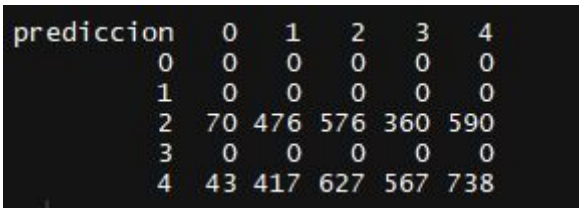
```
porcentaje<-0.7
set.seed(123)

corte <- sample(nrow(datos), nrow(datos)*porcentaje)
train <- datos[corte,]
test <- datos[-corte,]
```

    b. El resultado obtenido con el modelo fue el siguiente:

```
Naive Bayes Classifier for Discrete Predictors

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

A-priori probabilities:
Y
         0          1          2          3          4
0.02820781 0.20866179 0.26916136 0.22148352 0.27248552

Conditional probabilities:
   Type
Y       [,1]      [,2]
  0 1.572391 0.4955668
  1 1.536641 0.4987692
  2 1.457657 0.4982918
  3 1.407376 0.4914512
  4 1.415824 0.4929494
```

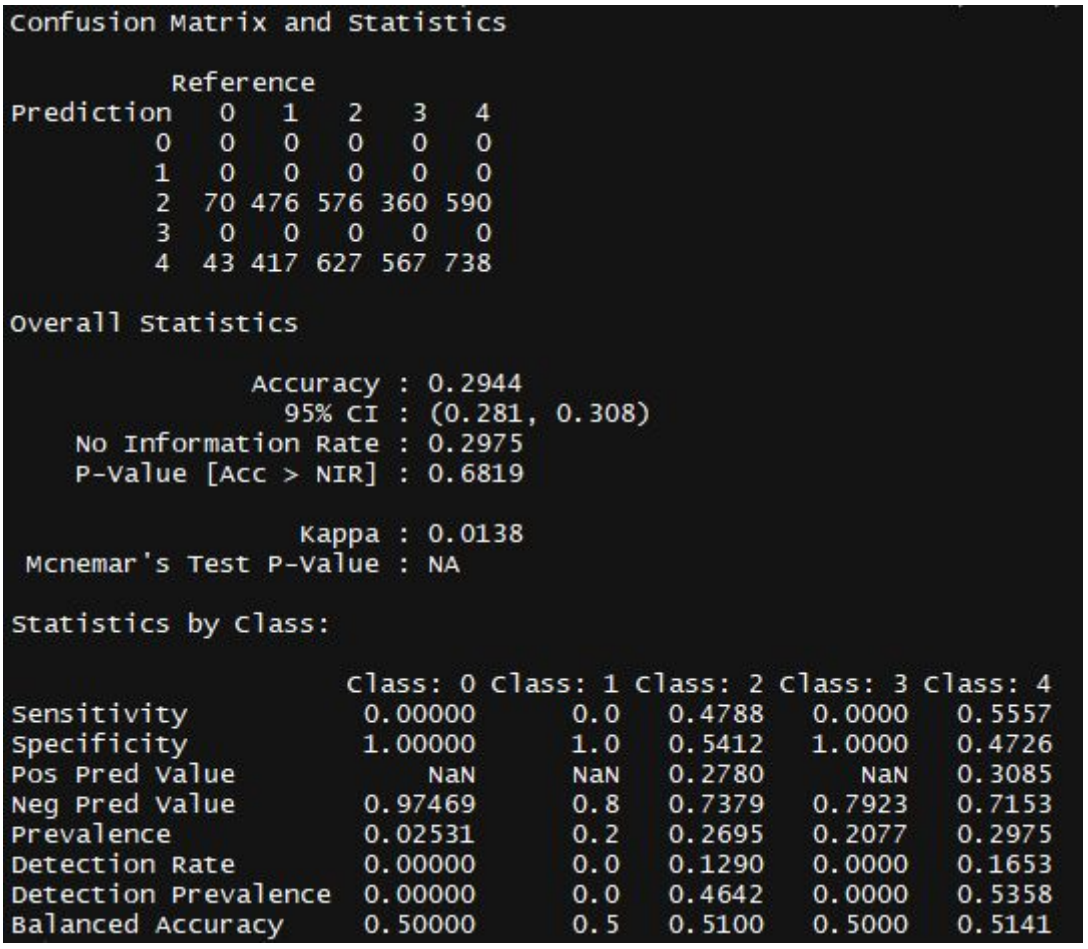
2. **Analizar el modelo y explicar si hay overfitting o no.**
    a. Por el resultado obtenido de la matriz de confusión, no hay overfitting ya que el accuracy obtenido no es muy bueno. Solo se obtuvo un 29.44% de accuracy (La imagen demostrando el porcentaje de accuracy se mostrará después).

3. **Utilizar el modelo con el conjunto de prueba y determinar la eficiencia del algoritmo para clasificar o predecir.**
    a. Como se puede observar en la siguiente imagen, el modelo no es tan eficiente para predecir. El modelo predijo que casi todas las probabilidades de adopción eran 2 o 4 cuando en realidad eran 0, 1, 2, 3 y 4.



prediccion	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	70	476	576	360	590
3	0	0	0	0	0
4	43	417	627	567	738

4. **Hacer una análisis de la eficiencia del algoritmo utilizando una matriz de confusión.**
    a. Al realizar la matriz de confusión para analizar la eficiencia del algoritmo de naive bayes para clasificar la velocidad de adopción de los animales según la base de datos, se puede observar que el algoritmo no es el mejor, pero tampoco el peor para predecir. Estos son los datos que nos indica la matriz de confusión:



Confusion Matrix and Statistics

Prediction \ Reference	0	1	2	3	4
0	0	0	0	0	0
1	0	0	0	0	0
2	70	476	576	360	590
3	0	0	0	0	0
4	43	417	627	567	738

Overall Statistics

Accuracy	0.2944
95% CI	(0.281, 0.308)
No Information Rate	0.2975
P-Value [Acc > NIR]	0.6819
Kappa	0.0138
Mcnemar's Test P-Value	NA

Statistics by Class:

	Class: 0	Class: 1	Class: 2	Class: 3	Class: 4
Sensitivity	0.00000	0.0	0.4788	0.0000	0.5557
Specificity	1.00000	1.0	0.5412	1.0000	0.4726
Pos Pred Value	NaN	NaN	0.2780	NaN	0.3085
Neg Pred Value	0.97469	0.8	0.7379	0.7923	0.7153
Prevalence	0.02531	0.2	0.2695	0.2077	0.2975
Detection Rate	0.00000	0.0	0.1290	0.0000	0.1653
Detection Prevalence	0.00000	0.0	0.4642	0.0000	0.5358
Balanced Accuracy	0.50000	0.5	0.5100	0.5000	0.5141

b. Como se pudo observar, se obtuvo un accuracy de 29.44% y un kappa de 0.0138. Esto significa que no se clasificaron perfectamente todas las velocidades de adopción, pero que aún se obtuvo un porcentaje mayor al 25% de exactitud. También significa que no es el mejor algoritmo a utilizar para clasificar o predecir este conjunto de datos.

5. **Comparar la eficiencia del algoritmo con los resultados obtenidos con el árbol de decisión y el modelo de regresión lineal. ¿Cual es el mejor para predecir? ¿Cual se demoró más en procesar?**
   a. El que más se ha tardado en procesar los datos ha sido el algoritmo de árbol de decisión (que nunca terminó de analizar los mismos demorándose más de 3 horas en hacerlo). Si se compara con los datos obtenidos anteriormente, el mejor ha sido el modelo de regresión lineal que obtuvo un accuracy de 55.51% contra el del modelo de naive bayes que obtuvo un accuracy de 29.44%. Es importante mencionar que estos dos no se demoraron mucho en procesar los datos.

```
Confusion Matrix and Statistics


      1    2
 1 1932  460
 2 1526  546

               Accuracy : 0.5551
                 95% CI : (0.5404, 0.5698)
    No Information Rate : 0.7746
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0737
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.5587
            Specificity : 0.5427
         Pos Pred Value : 0.8077
         Neg Pred Value : 0.2635
             Prevalence : 0.7746
         Detection Rate : 0.4328
   Detection Prevalence : 0.5358
      Balanced Accuracy : 0.5507

       'Positive' Class : 1
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1   2   3   4
         0   0   0   0   0   0
         1   0   0   0   0   0
         2  70 476 576 360 590
         3   0   0   0   0   0
         4  43 417 627 567 738

Overall Statistics

               Accuracy : 0.2944
                 95% CI : (0.281, 0.308)
    No Information Rate : 0.2975
    P-Value [Acc > NIR] : 0.6819

                  Kappa : 0.0138
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 0 Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity           0.00000      0.0   0.4788   0.0000   0.5557
Specificity           1.00000      1.0   0.5412   1.0000   0.4726
Pos Pred Value            NaN      NaN   0.2780      NaN   0.3085
Neg Pred Value        0.97469      0.8   0.7379   0.7923   0.7153
Prevalence            0.02531      0.2   0.2695   0.2077   0.2975
Detection Rate        0.00000      0.0   0.1290   0.0000   0.1653
Detection Prevalence  0.00000      0.0   0.4642   0.0000   0.5358
Balanced Accuracy     0.50000      0.5   0.5100   0.5000   0.5141
```

6. **Actualizar el kernel de kaggle:**
   Link al kernel: https://www.kaggle.com/rec16005/petterinosnaivebayes