Asignatura	Datos del alumno	Fecha
Aprendizaje Automático	Apellidos: FERNÁNDEZ DE LA TORRE	23-04-2019
	Nombre: CARLOS	

Trabajo: Clasificación con NaiveBayes

Descripción del trabajo

Mediante este trabajo se pretende que pongas en práctica la creación de modelos basados en el clasificador NaiveBayes. El objetivo es que comprendas de forma práctica con un problema determinado los pasos que hay que realizar para construir un clasificador NaiveBayes que infiera la clase más probable de dos eventos.

Para este trabajo puedes utilizar la herramienta **R** y el entorno de desarrollo **RStudio**, por tanto, debes tener correctamente descargado e instalado estos programas en tu ordenador.

- El intérprete del lenguaje R, lo puedes descargar desde aquí:https://cran.r-project.org/mirrors.html
- El programa RStudio lo puedes descargar desde aquí:https://www.rstudio.com/products/rstudio/download/#download

Además, puedes usar el paquete e1017 de R para la implementación del NaiveBayes: https://cran.r-project.org/web/packages/e1071/e1071.pdfy el paquete tm parallevar a cabo operaciones básicas de preparación de texto: https://cran.r-project.org/web/packages/tm/tm.pdf

Elaboración del trabajo

Durante este trabajo utilizarás los paquetes anteriores para realizar un clasificador NaiveBayes. Para ello es necesario procesar los datos de entrada (entrenamiento)

que serán proporcionados para el trabajo y entrenar los modelos correspondientes para llevar a cabo las predicciones sobre otros conjuntos de test.

Entrega del trabajo

Tras la realización del trabajo deberás entregar por un lado un fichero con el código que has realizado para generar los modelos y las predicciones y un informe que contenga los resultados obtenidos y una explicación del modelo desarrollado.

El informe tendrá una extensión máxima de 3 páginas siendo la fuente utilizada Georgia 11 e interlineado, 1,5.

Evaluación

El criterio de evaluación de esta actividad será tanto la capacidad predictiva obtenida en el conjunto de test, como la creatividad y rigurosidad llevada a cabo para obtener la solución.

Asignatura	Datos del alumno	Fecha
Aprendizaje Automático	Apellidos: FERNÁNDEZ DE LA TORRE	23-04-2019
	Nombre: CARLOS	

INFORME

Resultados obtenidos:

En primer lugar cargamos el conjunto de datos del archivo `house-votes-84.data`. Usamos `read.csv` siendo la coma el separador.

El documento contiene 435 instancias de las cuales 61.37931% son votantes demócratas y 38.62069% son republicanos.

Dado que el archivo `house-votes-84.data` no tiene en su primera fila los nombres de los campos, ponemos `header=False` en la función `read.csv` para que le añada al data frame la primera fila, posteriormente renombramos la **primera fila** para que el primer campo sea NAME según se ve en la siguiente figura:

```
> head(vote_data)
```

```
        NAME V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16
1 republican  n  y  n  y  y  y  n  n  n   y   ?   y   y   y   n   y
2 republican  n  y  n  y  y  y  n  n  n   n   n   y   y   y   n   ?
3   democrat  ?  y  y  ?  y  y  n  n  n   n   y   n   y   y   n   n
4   democrat  n  y  y  n  ?  y  n  n  n   n   y   n   y   n   n   y
5   democrat  y  y  y  n  y  y  n  n  n   n   y   ?   y   y   y   y
6   democrat  n  y  y  n  y  y  n  n  n   n   n   n   y   y   y   y
```

Observamos que hay datos con desconocidos **?** así que eliminamos los ? por datos vacíos.

```
str(vote_data)
```

```
'data.frame':	435 obs. of  17 variables:
 $ NAME: chr  "republican" "republican" "democrat" "democrat" ...
 $ V1  : chr  "n" "n" "" "n" ...
 $ V2  : chr  "y" "y" "y" "y" ...
 $ V3  : chr  "n" "n" "y" "y" ...
 $ V4  : chr  "y" "y" "" "n" ...
 $ V5  : chr  "y" "y" "y" "" ...
 $ V6  : chr  "y" "y" "y" "y" ...
 $ V7  : chr  "n" "n" "n" "n" ...
 $ V8  : chr  "n" "n" "n" "n" ...
 $ V9  : chr  "n" "n" "n" "n" ...
 $ V10 : chr  "y" "n" "n" "n" ...
 $ V11 : chr  "" "n" "y" "y" ...
 $ V12 : chr  "y" "y" "n" "n" ...
 $ V13 : chr  "y" "y" "y" "y" ...
 $ V14 : chr  "y" "y" "y" "n" ...
 $ V15 : chr  "n" "n" "n" "n" ...
 $ V16 : chr  "y" "" "n" "y" ...
```

Hacemos que la columna NAME sea una **variable categórica** pudiendo tomar únicamente los valores `democrat o republican.` Vemos cuantas instancias tenemos de cada clase:

```
> prop.table(table(vote_data$NAME))
```

```
  democrat republican
 0.6137931  0.3862069
```

Separamos los data sets de entrenamiento y de test observando que se mantienen las proporciones al menos hasta el primer decimal:

```
> vote_raw_train <- vote_data[1:370, ]
> vote_raw_test  <- vote_data[371:435, ]
> # Observamos que se mantienen las proporciones
> prop.table(table(vote_raw_train$NAME))
```

```
  democrat republican
 0.6108108  0.3891892
```

```
> prop.table(table(vote_raw_test$NAME))

   democrat republican
 0.6307692  0.3692308
```

Entrenamos un clasificador NB.

El modelo utiliza la presencia "yes" o ausencia "no" de uno de los 16 restantes atributos para estimar la probabilidad de que un votante sea demócrata o republicano.

`laplace=1` es necesario incorporar una pequeña corrección de muestreo para evitar que la probabilidad sea cero.

```
> vote_classifier <- naiveBayes(vote_raw_train, vote_raw_train$NAME, laplace = 1)
```

Predecimos la clase más probable con `predict` donde `vote_classifier` es el modelo entrenado y `vote_raw_test` los datos para testear el modelo.

El array `vote_test_pred` contiene la predicción para los 65 casos de test.

```
> vote_test_pred <- predict(vote_classifier, vote_raw_test, type = "class")
```

Matriz de confusión:

```
> table(vote_test_pred, vote_raw_test$NAME)

vote_test_pred democrat republican
    democrat         34          1
    republican        7         23


> prop.table(table(vote_test_pred, vote_raw_test$NAME), margin = 2)

vote_test_pred   democrat republican
    democrat   0.82926829 0.04166667
    republican 0.17073171 0.95833333
```

Predicciones donde se ha equivocado el modelo:

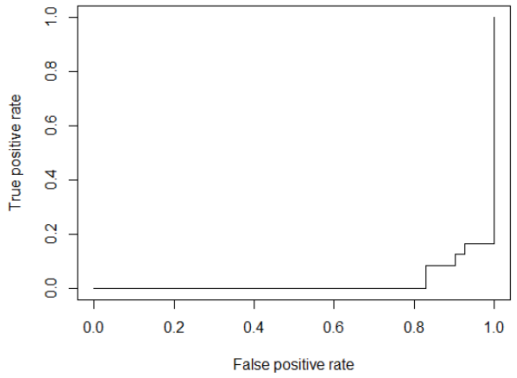
```
> vote_raw_test[vote_raw_test$NAME != vote_test_pred,] # Comparamos los datos de las
dos columnas

          NAME V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16
373   democrat  n  y  y  y  y  y  n  n  n   n   y   y       y   n   n
376   democrat  n  y  n  y  y  y  n  n  n   n   y   y   n   y   n   n
383   democrat  y  y  n  y  y  y  n  n  n   n   y   n   y   y   n
385   democrat  y  y  y  y  y  y  n  n  n   n   y   y   y   y   n   y
386   democrat  y  y  n  n  y  y  n  n  n   n   y   y   y   y   y   n
389   democrat  n  y  y  y  y  y  n  n  n   n   n   y   y   y   n
394 republican                n  y  n  y  y   n   n   y   y   n   n
408   democrat  n  n  n  y  y  y  n  n  n   n   y   y   y   y   n   n
```

Predecimos con probabilidades:

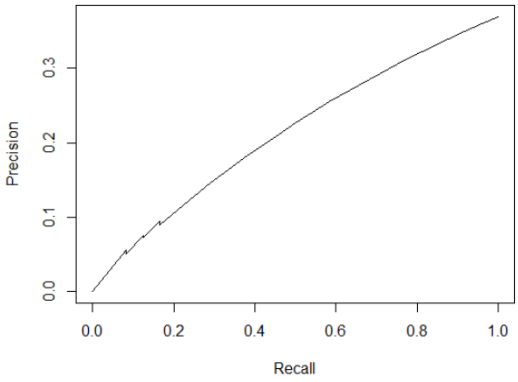
```
> vote_test_pred <- predict(vote_classifier, vote_raw_test, type = "raw")
> pred <- as.data.frame(vote_test_pred)
> head(pred)

      democrat   republican
1 1.000000e+00 1.737201e-09
2 1.000000e+00 8.871248e-13
3 2.481850e-04 9.997518e-01
4 9.419610e-01 5.803897e-02
5 6.960115e-11 1.000000e+00
6 1.921005e-05 9.999808e-01
```

Finalmente representamos las distintas gráficas `Curva ROC`, `Curva Precision-Recall` y `Curva sensivity-specificity`.

`Curva ROC`



`Curva Precision-Recall`



`Curva sensivity-specificity`