Proyecto final

Jiménez Díaz de Sandi René

Instituto Tecnológico de Tijuana, Departamento de Sistemas y Computación, Tijuana, Baja California, México.

rene.jimenez17@tectijuana.edu.mx

Resumen. **- Proyecto final de la materia de Estadística y Análisis de Datos utilizando el algoritmo Naive Bayes y el lenguaje de programación estadística R.**

I. INTRODUCCION

En este proyecto se pretende averiguar la cobertura del sistema de recolección de basura con base a los resultados del Censo de 2015 en el estado de Baja California realizado por el Instituto Nacional de Estadística y Geografía (INEGI).

II. MARCO TEÓRICO

R

Sistema para computación estadística y gráficos. Consiste en un lenguaje más un entorno de tiempo de ejecución con gráficos, un depurador, acceso a ciertas funciones del sistema y la capacidad de ejecutar programas almacenados en archivos de script. El núcleo de R es un lenguaje informático interpretado que permite la ramificación y el bucle, así como la programación modular mediante funciones. La mayoría de las funciones visibles para el usuario en R están escritas en R. Es posible que el usuario interactúe con los procedimientos escritos en los lenguajes C, C ++ o FORTRAN para mayor eficiencia. La distribución R contiene funcionalidad para una gran cantidad de procedimientos estadísticos. Entre estos se encuentran: modelos lineales y lineales generalizados, modelos de regresión no lineal, análisis de series de tiempo, pruebas paramétricas y no paramétricas clásicas, agrupamiento y suavizado. También hay un gran conjunto de funciones que proporcionan un entorno gráfico flexible para crear varios tipos de presentaciones de datos. Los módulos adicionales ("paquetes de complementos") están disponibles para una variedad de propósitos específicos (consulte R Paquetes de complementos) [1].

Censo

El Censo de Población y Vivienda tiene como objetivo principal producir la cuenta de la población residente del país, así como la información sobre su estructura y principales características socioeconómicas y culturales, además de su distribución en el territorio nacional; del mismo modo obtener la cuenta del total de viviendas y sus características, sin perder, en la medida de lo posible, la comparabilidad histórica a nivel nacional e internacional [2].

Análisis de datos

El análisis de datos consiste en la realización de las operaciones a las que el investigador someterá los datos con la finalidad de alcanzar los objetivos del estudio. Todas estas operaciones no pueden definirse de antemano de manera rígida. La recolección de datos y ciertos análisis preliminares pueden revelar problemas y dificultades que desactualizarán la planificación inicial del análisis de los datos. Sin embargo, es importante planificar los principales aspectos del plan de análisis en función de la verificación de cada una de las hipótesis formuladas ya que estas definiciones condicionarán a su vez la fase de recolección de datos [3].

Algoritmo clasificador de Naive Bayes

Los clasificadores ingenuos de Bayes son una colección de algoritmos de clasificación basados en el teorema de Bayes. No es un algoritmo único, sino una familia de algoritmos en los que todos ellos comparten un principio común, es decir, cada par de características que se clasifican es independiente entre sí [4]. El teorema de Bayes se compone de la siguiente formula:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Usando el teorema de Bayes, podemos encontrar la probabilidad de que ocurra A, dado que B ha ocurrido. Aquí, B es la evidencia y A es la hipótesis. La suposición hecha aquí es que los predictores / características son independientes. Es decir, la presencia de una característica particular no afecta a la otra [5].

Ggplot2

El paquete ggplot2, creado por Hadley Wickham, ofrece un lenguaje gráfico poderoso para crear tramas elegantes y complejas. Su popularidad en la comunidad R ha explotado en los últimos años. Originalmente basado en The Grammar of Graphics de Leland Wilkinson, ggplot2 le permite crear gráficos que representan datos numéricos y categóricos univariados y multivariados de una manera directa. La agrupación se puede representar por color, símbolo, tamaño y transparencia. La creación de gráficos enrejados (es decir, acondicionamiento) es relativamente simple [6].

E1071

Funciones para análisis de clase latente, transformadas de Fourier de corto tiempo, agrupación difusa, máquinas de vectores de soporte, cálculo de ruta más corta, agrupación en bolsas, clasificador bayesiano, entre otros [7].

Caret

El paquete caret (abreviatura de Entrenamiento de Clasificación y Regresión) contiene funciones para simplificar el proceso de entrenamiento modelo para problemas complejos de regresión y clasificación. El paquete utiliza una cantidad de paquetes R pero intenta no cargarlos todos al inicio del paquete (al eliminar las dependencias formales del paquete, el tiempo de inicio del paquete puede disminuir considerablemente). El campo paquete "sugiere" incluye 30 paquetes. caret carga los paquetes según sea necesario y asume que están instalados. Si falta un paquete de modelado, hay un mensaje para instalarlo [8].

III. DESARROLLO

- Se declaran las librerías a utilizar

```
library(Amelia)
library(e1071)
library(caret)
library(caTools)
```

- Se almacena los datos del archivo TR_VIVIENDA02.csv en la variable “ep”

```
ep <- read.csv('C:/Users/rjds_/Downloads/TR_VIVIENDA02.csv')
```

- Declaramos una semilla de 100

```
set.seed(100)
```

- Se selecciona las columnas a utilizar

```
vivienda <- ep[c(4,40,88)]
```

- Restringimos a un ingreso menor de $100,000 ya que mayores a esa cantidad usan de manera correcta los servicios de recolección de basura

```
vivienda <- vivienda[vivienda$INGTRHOG < 100000,]
```

- Se realiza un proceso de limpieza sobre los datos a utilizar

```
missmap(vivienda)
```

- Se valida que los datos no contengan datos nulos

```
vivienda <- na.omit(vivienda)
```

- Se realiza un segundo proceso de limpieza con la finalidad de evitar errores al momento de entrenar nuestro modelo

```
missmap(vivienda)
```

- Se indica que usaremos el campo “destino basura” para crear nuestras clases

```
vivienda$DESTINO_BASURA = factor(vivienda$DESTINO_BASURA, levels = c(1,2,3,4,5,6))
```

- Indicamos que nuestro modelo usara un 70% de datos de entrenamiento y un 30% de prueba

```
split = sample.split(vivienda$DESTINO_BASURA, SplitRatio = 0.7)
```

- Se declaran los datos de entrenamiento

```
traindata = subset(vivienda, split == TRUE)
```

- Declaramos los datos de prueba

```
testdata = subset(vivienda, split == FALSE)
```

- Se elimina la variable donde almacenaba los datos

```
rm(split)
```

- Entrenamos nuestro modelo bayesiano

```
nbmodel <- naiveBayes(x = traindata, y = traindata$DESTINO_BASURA, SplitRatio = 0.70)
```

- Comprobamos que no contenga errores

```
nbmodel
```

- Indicamos nuestros datos de predicción

```
pred <- predict(nbmodel,testdata[,-3])
tab <- table(testdata[,2], pred, dnn = c("Actual", "Predicted"))
```

- Mostramos nuestro modelo de predicción

```
confusionMatrix(tab)
```

- Limpiamos la memoria

```
gc()
```

- Se muestra de manera gráfica nuestros datos de entrenamiento

```
plt  <- ggplot(traindata, aes(x=MUN, y=INGTRHOG, color = traindata$DESTINO_BASURA))
```

- Utilizando la librería ggplot para mostrar la gráfica, vamos a utilizar el subset de los datos de entrenamiento del modelo creados en el split, a continuación, colocamos los datos de las columnas a mostrar en el eje X como en el eje Y, se van a mostrar dependiendo el destino de la basura.
-

```
+ labs(title = 'Ingresos del hogar por municipio', x="Municipios",y="Ingresos x hogar", color='Destino de la basura' )
```

- A continuación, se va a utilizar la función para el grafico de puntos más la estructura anterior creada.

```
trainplt <- plt + geom_jitter()
//Impresión de la Grafica.
trainplt
```

- Se muestra de manera gráfica nuestros datos de prueba

```
plat  <- ggplot(testdata, aes(x= MUN, y= INGTRHOG,color = testdata$DESTINO_BASURA))
//utilizando la librería ggplot para mostrar la gráfica, vamos a utilizar el subset de los datos de prueba del modelo creados en el split, a continuación, colocamos los datos de las columnas a mostrar en el eje X como en el eje Y, se van a mostrar dependiendo el destino de la basura
//Utilizando labs vamos a colocar el nombre de los labels a los ejes y a la grafica

+ labs(title ='Ingresos del hogar por municipio', x="Municipios",y="Ingresos x hogar", color='Destino de la basura' )
testplt <- plat + geom_jitter()
Testplt
```

IV. INTERPRETACIÓN DE RESULTADOS

En las siguientes graficas se presentan 6 niveles/tipos de destino de basura, donde:

TABLA 1

Valor	Concepto
1	Se entrega a un camión de basura
2	Se deposita en un contenedor de basura
3	Se quema la basura
4	Se entierra la basura
5	Se lleva a un basurero publico
6	Se tira en otro lugar

Mientras que para los municipios se utilizan los siguientes valores:

TABLA 2

Valor	Concepto
1	Ensenada
2	Mexicali
3	Tecate
4	Tijuana
5	Playas de Rosarito

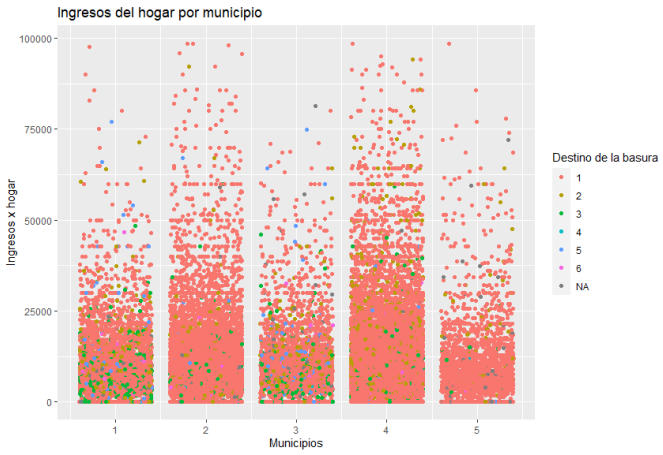

Fig.1 Resultado con el set de entrenamiento del algoritmo.

En la gráfica podemos observar que el destino de basura principal es el número 1, el cual abarca la gran mayoría de los puntos, de los 5 municipios el que podemos apreciar que tiene más ingresos por hogar es el número 4 que pertenece a Tijuana y el que menor ingreso por hogar tiene es el de Playas de Rosarito donde se observa una gran concentración de puntos debajo del ingreso mensual de $25,000.

Fig.2 Resultado con el set de prueba del algoritmo.

En el caso de los entrenamientos de prueba podemos observar que la tendencia es muy similar al caso anterior, donde podemos observar que Tijuana es de los municipios con mayor ingreso por hogar frente a Playas de Rosarito es el que menos ingreso por hogar tiene, pero podemos ver que el número 1 que pertenece a Ensenada tiene el punto más alto de ingresos frente los demás municipios.

Dejando de lado la interpretación de las gráficas individuales y dando una interpretación en conjunto, estas nos indican que el sistema de recolección de basura es deficiente en colonias o asentamientos de menor ingreso mensual lo que puede llegar a generar condiciones insalubres y de marginación aun en localidades próximas o dentro de ciudades.

V. CONCLUSION

El clustering es una herramienta muy útil para agrupar datos similares ya que su uso va desde la separación de verduras por campesinos hasta su aplicación en aprendizaje no supervisado en inteligencia artificial, en este caso lo utilizamos para agrupar datos del CSV de vivienda en el cual agrupamos datos de los municipios vs el ingreso por hogar y con esto tuvimos los datos para hacer el análisis para saber qué municipio tiene más este servicio según su rango de ingreso en sus hogares. En este caso, al aplicar el algoritmo Naive Bayes notamos que es bastante simple de aplicar, pero puede ofrecer predicciones incorrectas ya que al tomar en cuenta las variables de forma independiente no toma en cuenta las situaciones del mundo real donde algunas variables se ven afectadas entre sí.

VI. BIBLIOGRAFIA

[1] Hornik, K. (20-02-20). R FAQ. R FAQ. https://cran.r-project.org/doc/FAQ/R-FAQ.html#What-is-R_003f

[2] INEGI. (s. f.). Censo Población y Vivienda 2020. Recuperado 30 de junio de 2020, de https://www.inegi.org.mx/programas/ccpv/2020/default.html

[3] ANALISIS DE DATOS - Técnicas de Investigación Educativa G38. (s. f.). Técnicas de Investigación Educativa G38. Recuperado 30 de junio de 2020, de https://sites.google.com/site/tecnicasdeinvestigaciond38/metodos-estadisticos/1-1-analisis-de-datos

[4] Geeks for Geeks. (s. f.). Naive Bayes Classifiers. Recuperado de https://www.geeksforgeeks.org/naive-bayes-classifiers/

[5] Gandhi, R. (2018, 5 mayo). Naive Bayes Classifier. Recuperado 6 de junio de 2020, de https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

[6] Kabacoff, R. (s. f.). ggplot2 Graphs. Quick - R. Recuperado 30 de junio de 2020, de https://www.statmethods.net/advgraphs/ggplot2.html

[7] e1071. (s. f.). Cran. Recuperado 30 de junio de 2020, de https://cran.r-project.org/web/packages/e1071/index.html+

[8] CRAN Project. (s. f.). A Short Introduction to the caret Package. CRAN. Recuperado 30 de junio de 2020, de https://cran.r-project.org/web/packages/caret/vignettes/caret.html