# Práctica 6

# Variables Dummies – KNN – Análisis Discriminante – Naïve Bayes

## 1. Variables Dummies (One-Hot Encoding)

En los problemas de datos, generalmente tenemos que trabajar con variables numéricas y variables categóricas. En el caso de las categóricas, si bien hay algoritmos que nos permiten trabajar con ellas, en la mayoría de los casos necesitamos que los datos sean representados de forma numérica. Para eso, utilizaremos las Variables Dummies o One Hot Encoding.

Es un método sencillo que consiste en crear una columna nueva, para cada valor distinto que toma la variable categórica que estamos queriendo utilizar y asignarle a cada observación un 1 cuando corresponde con el nivel de la categoría y un 0 en otro caso. En la siguiente tabla se muestra un ejemplo para la variable “sexo”, que tiene dos niveles: “hombre” o “mujer”:

| Sexo | Sexo_hombre | Sexo_mujer |
|---|---|---|
| Hombre | 1 | 0 |
| Hombre | 1 | 0 |
| Mujer | 0 | 1 |
| Hombre | 1 | 0 |
| Mujer | 0 | 1 |
| Mujer | 0 | 1 |
| Hombre | 1 | 0 |
| Mujer | 0 | 1 |
| Hombre | 1 | 0 |

### Implementación en R

En R podemos utilizar la función *dummy_cols()* del paquete **fastDummies** [2]. Los parámetros que vamos a utilizar son:

- .data: el objeto de datos del que vamos a generar las columnas dummies.
- select_columns: vector con los nombres de las columnas de las que vamos a crear las variables dummies.
- remove_selected_columns: si es verdadero, elimina las columnas usadas para generar las columnas dummies.

## 2. K-vecinos más cercanos

K-vecinos más cercanos (k-NN por sus siglas en inglés), es un algoritmo de clasificación supervisada que estima la probabilidad de que un elemento de un conjunto de datos pertenezca a una clase, de acuerdo con las características de un conjunto de datos tomados como ejemplo (conjunto de entrenamiento). No hace ninguna suposición sobre las variables predictoras.

A un elemento se le asigna una clase, si es la más frecuente entre los k elementos más cercanos. Esa "cercanía" se establece con la distancia euclidiana entre esos elementos [1].

El valor de k (n° de vecinos) afectará el resultado de la clasificación, por lo que es conveniente generar varios modelos con distintos valores para obtener mejores resultados de clasificación.

### Estimación con R

Para encontrar un modelo de clasificación por k-NN, vamos a utilizar la función del paquete **class** de la instalación base de R: *knn()*.

## 3. Análisis discriminante

El análisis discriminante lineal (LDA por sus siglas en inglés) es un método de clasificación supervisado, que se basa en la obtención de funciones lineales a partir de las variables predictoras, que permitan diferenciar los grupos de la variable dependiente (utilizando el teorema de Bayes).

Si la separación entre las clases de la variable dependiente no es suficiente y están muy solapados los elementos, no será posible encontrar una función discriminante que me permita clasificar correctamente los datos.

En el caso del análisis discriminante cuadrático (QDL), es similar al LDA, pero no se basa en funciones lineales sino en funciones cuadráticas y por esto los límites de decisión entre los grupos son curvos. Esto permite la clasificación cuando no hay una separación lineal entre las clases.

### Estimación con R

El paquete **MASS** tiene dos funciones para encontrar modelos discriminante lineal y cuadrático: *lda()* y *qda()* [3].

Los parámetros que se van a utilizar, en los dos casos, son:

- formula: vamos a indicar que variable va a ser la dependiente y cuáles las explicativas, con el operador **~**.
- data: el data.frame del que se especifican las variables (opcional).

## 4. Clasificador Bayesiano

El clasificador naïve Bayes o Bayes ingenuo, es muy utilizado para problemas de clasificación, en donde las variables explicativas son categóricas. Esto se debe a que es rápido y simple. La denominación de "ingenuo" es porque supone que las variables explicativas son independientes.

Se basa en el teorema de probabilidad condicional o teorema de Bayes, cuya expresión general es:

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

Calcula la probabilidad de A, condicionado a B (a que ha ocurrido B). Dónde:

- P(A): probabilidad de A,
- P(B): probabilidad de B y es > 0;
- P(B|A): probabilidad de B dado A.

**Implementación en R**

Para implementar un clasificador bayesiano en R, utilizaremos dos funciones: *naiveBayes()* del paquete **e1071** [4] y *naive_bayes()* del paquete **naivebayes** [5].

Los parámetros que vamos a utilizar de ambas funciones son:

- formula: vamos a indicar que variable va a ser la dependiente y cuáles las explicativas, con el operador **~**.
- data: el data.frame del que se especifican las variables (opcional).

Para este caso de clasificación, podemos hacer validación mediante k-fold cross validation, utilizando la función *tune()* del paquete **e1071**. La forma de implementarla es muy simple: hay que definir dos tipos de parámetros, por un lado, los parámetros de control de la validación y por el otro, los parámetros del modelo que utilizaremos.

- tunecontrol: mediante la función tune.control(), de la que a su vez vamos a utilizar **sampling** y **cross**.
- method: la función que vamos a ajustar con tune, en este caso **naiveBayes**.
- train.x: ídem a formula para este caso.
- data: el data.frame del que se especifican las variables (opcional).

Una vez ejecutado, el modelo resultante lo obtenemos del objeto **$best.model** resultante de ejecutar *tune()*.

## 5. Evaluación de métricas de clasificación con R

En la teoría se vieron las distintas métricas para evaluar “que tan bueno” es un modelo de clasificación, derivadas de la matriz de confusión.

Para generar la matriz de confusión del modelo y encontrar las métricas, vamos a utilizar la función *confusionMatrix()* del paquete **caret** [6]. Donde vamos a usar los siguientes parámetros:

- data: un vector de factores que es el resultado de nuestra predicción.

reference: un vector de factores que tiene las verdaderas clases del conjunto de datos de testeo.

**Ejercicios**

**Ejercicio 1 –** Utilice los datos de la UNER, con los que encontró un modelo de regresión logística en el ejercicio 3 de la guía anterior, para encontrar un nuevo modelo para clasificar si el estudiante asistió a clases de consulta utilizando todas las variables disponibles de forma correcta.

Evalúe el modelo encontrado y encuentre el porcentaje de datos bien clasificados. Compare el resultado con el modelo de la guía anterior.

**Ejercicio 2** – En el archivo "demencia.xls" se presentan datos de un estudio sobre los factores de riesgo asociados con el Alzheimer. Se quiere determinar si los incidentes de demencia pueden relacionarse con el consumo de vino y otras variables. El estudio se realizó sobre una muestra de adultos mayores entre los cuales hay algunos sin la enfermedad.

Las variables incluidas son:

ID: código de identificación.
AGE: edad en años.
WINE: consumo de vino (0 = no consume; 1 = hasta ¼ l diario; 2 = más de ¼ l diario).
MMSE: mini-mental, examen de estado mental, con puntaje de 0 a 30.
HIGHBP: presión diastólica alta (1 = sí; 0 = no).
T3DEMEN: incidentes de demencia (1 = sí; 0 = no).

a. Explore los datos.

b. Ajuste un modelo de regresión logística que permita predecir posibles incidentes de demencia, utilizando adecuadamente todas las variables medidas.

c. ¿Qué variables no son relevantes en el modelo?

d. ¿Cuál es el porcentaje de casos bien clasificados?

e. De acuerdo con el modelo ajustado, ¿cuál es la probabilidad que le asigna a un nuevo caso con los siguientes datos: AGE= 70; WINE= 1; MMSE= 25; HIGHBP= 1? ¿A qué grupo pertenecería?

**Ejercicio 3** - Utilice el dataset "iris" de R, separe en datos de entrenamiento y testeo (80% train y 20% test) y con el algoritmo k-NN entrene un modelo de clasificación, utilizando un número k de vecinos de 5. Conteste:

a. Encuentre la matriz de confusión y las métricas de la predicción.

b. Encuentre el error medio de cada clasificación.

c. ¿Cómo supone que puede mejorar el resultado?

**Ejercicio 4** – En el archivo “glass.csv” se presentan datos del Servicio de Ciencias Forenses de los Estados Unidos sobre 6 tipos de vidrio, definidos de acuerdo con su contenido de óxido de varios elementos [7].

a. Explore los datos.

b. Divida en un conjunto de entrenamiento y de testeo con la relación 70% - 30%.

c. Determine el número de vecinos adecuado y entrene un modelo para clasificar los tipos de vidrios.

Encuentre las métricas del modelo obtenido.

**Ejercicio 5** – Se tienen los valores estadísticos relacionados con los intervalos RR de señales electrocardiográficas de 8528 pacientes (datasetRR.csv).

Los pacientes pueden presentar un electrocardiograma normal (N), con fibrilación auricular (A), ruidoso (~) o con otros ritmos cardíacos (O).

a. Explore los datos. ¿Qué proporción de los datos tiene cada clase?

b. Separe el conjunto de datos en 90% de entrenamiento y 10% de testeo.

c. Entrene un modelo utilizando el algoritmo k-NN y evalúe la clasificación en los datos de prueba.

d. Compare la clasificación obtenida con un modelo LDA y QDA. ¿Cuál tiene mejor desempeño?

**Ejercicio 6** – Se tienen los datos relacionados con la supervivencia de 306 pacientes a la operación de cáncer de mama (haberman.csv) [8].

Encuentre un modelo utilizando LDA y QDA para predecir la supervivencia del paciente de acuerdo con:

a. Explore los datos. ¿Utilizaría todas las variables para generar el modelo?

b. Utilice los métodos de validación simple y LOOCV para generar el modelo y evalúe su desempeño con la métrica AUC.

c. Entrene un modelo utilizando validación k-fold CV y evalúe la clasificación. ¿Mejoró respecto a los otros métodos de validación?

**Ejercicio 7** – Cargue los datos del archivo “titanic.csv” y encuentre un modelo utilizando el algoritmo de clasificación bayesiana para predecir la supervivencia o no, de los pasajeros del Titanic.

a. Explore los datos. ¿Qué variables utilizaría como explicativas del modelo?

b. Realice un modelo bayesiano utilizando validación simple (80%-20%).

c. Evalúe el modelo.

**Ejercicio 8** – A partir de los datos contenidos en el archivo “trenes.txt”, utilice un clasificador bayesiano para determinar la puntualidad de los trenes.

a. Explore los datos.

b. Encuentre un modelo con los datos indicados utilizando validación simple. Evalúe el modelo.

c. Repita el ítem anterior para los datos del archivo “trenes2.txt”. ¿Qué resultado obtiene?

¿Cómo puede mejorar la clasificación del modelo?

**Referencias**

1. “Estadística y Machine Learning con R”. Disponible en: https://bookdown.org/content/2274/portada.html
2. “Package fastDummies”. Disponible en: https://cran.r-project.org/web/packages/fastDummies/fastDummies.pdf
3. “Package MASS”. Disponible en: https://cran.r-project.org/web/packages/MASS/MASS.pdf
4. “Package e1701”. Disponible en: https://cran.r-project.org/web/packages/e1071/e1071.pdf
5. “Package naivebayes”. Disponible en: https://cran.r-project.org/web/packages/naivebayes/naivebayes.pdf
6. “Package caret”. Disponible en: https://cran.r-project.org/web/packages/caret/caret.pdf
7. “Glass Identification Data Set”, disponible en: https://archive.ics.uci.edu/ml/datasets/glass+identification
8. “Haberman´s Survival”. Disponible en: https://archive-beta.ics.uci.edu/ml/datasets/haberman+s+survival