

practica_final_laura_moreno

Vamos a utilizar el dataset de semillas que se encuentra aquí:

<https://archive.ics.uci.edu/ml/datasets/seeds#>

Primero vamos a descargarnos el dataset con el siguiente comando:

```
library(tidyverse)

— Attaching core tidyverse packages ————— tidyverse 2.0.0 —
✓ dplyr     1.1.4     ✓ readr     2.1.5
✓ forcats   1.0.0     ✓ stringr   1.5.1
✓ ggplot2   3.5.1     ✓ tibble    3.2.1
✓ lubridate 1.9.3     ✓ tidyr    1.3.1
✓ purrr    1.0.2

— Conflicts ————— tidyverse_conflicts() —
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()   masks stats::lag()

i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
df_seeds <- read.table('https://archive.ics.uci.edu/ml/machine-learning-databases/00236/seeds_da...
```

PREGUNTA 1 (0.5pt)

¿Cuantas filas y cuantas columnas tiene el dataframe df_seeds?

Respuesta:

```
# Ver número de filas y columnas
num_filas <- nrow(df_seeds)
num_columnas <- ncol(df_seeds)

num_filas

[1] 210

num_columnas

[1] 8
```

PREGUNTA 2 (0.5pt)

Vamos a convertir en factor la columna tipo. Vamos a reemplazar los números por su correspondiente etiqueta (label). La correspondencia entre el código y el tipo es:

- 1 - Kama
- 2 - Rosa
- 3 - Canadian

Convierte en factor la columna tipo, respetando las etiquetas:

Respuesta:

```
# Definir etiquetas
etiquetas <- c("Kama", "Rosa", "Canadian")

# Asignación etiquetas
df_seeds$tipo <- factor(df_seeds$tipo, levels = 1:3, labels = etiquetas)
```

PREGUNTA 3 (1pt)

¿Cuál es la media del area de cada uno de los tipos?

Respuesta

```
# Media del área por tipo
media_area <- aggregate(area ~ tipo, data = df_seeds, FUN = mean)

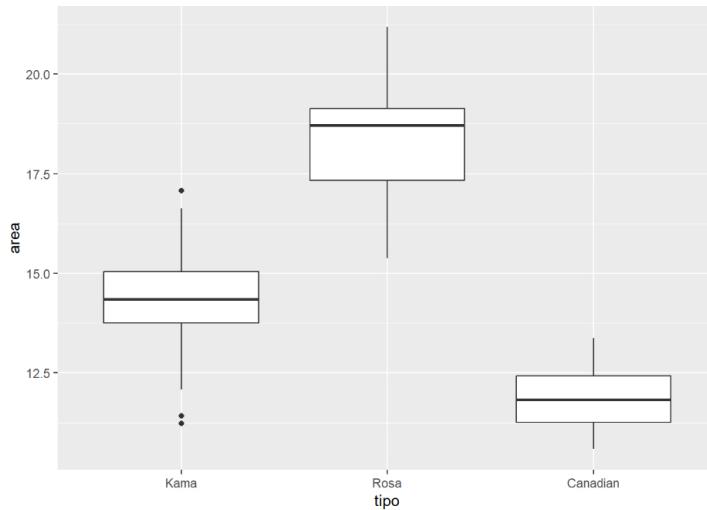
media_area

  tipo      area
1 Kama 14.33443
2 Rosa 18.33429
3 Canadian 11.87386
```

PREGUNTA 4 (0.5pt)

¿Como se llama el siguiente tipo de gráfico?. ¿Qué representa la línea del centro de la caja?

```
ggplot(df_seeds, aes(x=tipo, y=area)) + geom_boxplot()
```



Respuesta: Es un Boxplot, un Diagrama de Cajas o de Cajas y Bigotes. La línea del centro de la caja representa la mediana de los datos (el valor central de los datos ordenados). La caja representa el rango intercuartílico (IQR), los bigotes (las líneas que quedan fuera) representan los valores variables que quedan fuera del rango intercuartílico y los puntos que se alejan de los bigotes representan los outliers (valores atípicos).

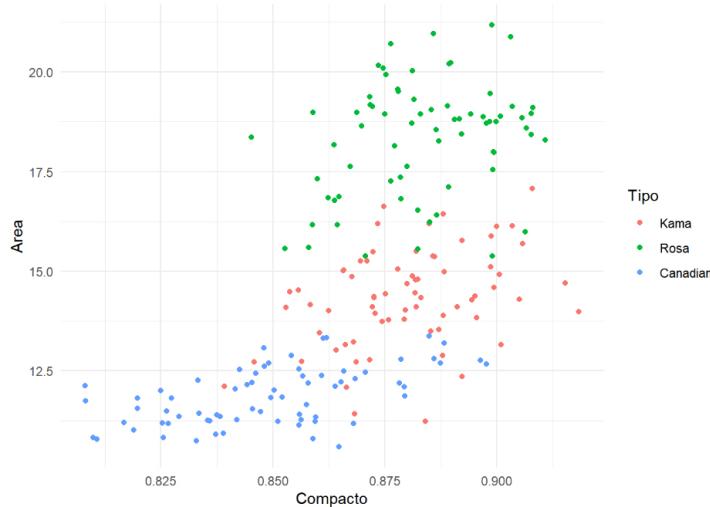
El rango intercuartílico es una medida de dispersión que se usa en estadística para comprender cómo se distribuyen los datos al rededor de la mediana, se emplea porque es menos sensible a los valores atípicos.

PREGUNTA 5 (1.5pt)

¿Como pintarías un diagrama de puntos (o scatterplot) con ggplot con las siguientes características? - En el eje X la variable compacto - En el eje Y la variable area - Cada tipo de semilla debería tener un color diferente

Respuesta:

```
ggplot(df_seeds, aes(x = compacto, y = area, color = tipo)) +
  geom_point() +
  labs(x = "Compacto", y = "Area", color = "Tipo") +
  theme_minimal()
```



PREGUNTA 6 (0.5pt)

¿Qué hace la siguiente línea?:

```
df_seeds |> mutate(is_kama = tipo=="Kama") -> df_seeds
```

Respuesta: Tras ejecutar esta línea, observamos que aparece una nueva columna en el conjunto de datos que se llama "is_kama" o "es_kama", a la cual se le asignan automáticamente valores lógicos booleanos (True y False, o Verdadero y Falso) que indican si la muestra pertenece o no al tipo "kama".

La función "mutate()" es la que se emplea para agregar la nueva columna, agregando a ella el nombre de la columna y la asignación de valor "true" si el tipo es igual a "kama".

PREGUNTA 7 (1.5pt)

Vamos a dividir el conjunto de datos en test y training porque vamos a entrenar un modelo que me permita diferenciar si una semilla es de tipo Kama o no. ¿Por qué es aconsejable dividir el dataset en los

grupos de train y test?

```
set.seed(123) # Este set.seed hace que a todos nos generen los mismos número aleatorios
idx <- sample(1:nrow(df_seeds), 0.7*nrow(df_seeds))
df_seeds_train <- df_seeds[idx,]
df_seeds_test <- df_seeds[-idx,]
```

Respuesta: La división de datos en conjuntos de entrenamiento y test es imprescindible a la hora de de entrenar modelos de aprendizaje automático porque nos permite evaluar la precisión que tiene el modelo para clasificar datos que no le han sido proporcionados anteriormente. Gracias a esta práctica podemos evitar errores como overfitting o underfitting del modelo (sobreajuste o subajuste) y realizar pruebas para comprobar la robustez y la precisión de modelos a la hora de predecir resultados.

Por ejemplo, si nuestro modelo tiene bajo rendimiento tanto en el conjunto de pruebas como en el de entrenamiento, puede estar indicándonos que el modelo es demasiado simple y tiene underfitting. También al separar en entrenamiento y test, podemos comparar el rendimiento del modelo en ambos conjuntos para saber si este no generaliza bien ante nuevos datos no proporcionado en dichos conjuntos, lo cual sería overfitting.

PREGUNTA 8 (1pt)

Vamos a crear un modelo para realizar una clasificación binaria, donde le pasaremos como entrada las columnas: area, perimetro, compacto, longitud, coeficient.asimetria y longitud.ranura

¿Qué tipo de algoritmo o modelo debería usar?

Respuesta: Para una tarea de clasificación binaria con datos como los que estamos manejando sería muy recomendable emplear la regresión logística para determinar la probabilidad de pertenencia o no a una clase determinada. Pero también existen otros modelos de clasificación dentro del aprendizaje supervisado que podrían ayudarnos con esta tarea, como máquinas de soporte vectoriales, árboles de decisión, bosques aleatorios o redes neuronales. Algunas pueden llegar a ser demasiado complejas de implementar para tareas sencillas de clasificación binarias como la que se establece en el supuesto.

Sé que no hemos profundizado al máximo en todos estos modelos, teniendo en cuenta el tiempo tan limitado con el que hemos contado para este módulo del curso, pero intento dar la respuesta más amplia posible según los conocimientos que he ido adquiriendo en los últimos meses. Llevo desde agosto del año pasado formándome en Python, Análisis de Datos y Machine Learning, y debo reconocer que me entusiasma el tema.

PREGUNTA 9 (1pt)

Crea un modelo que me permita clasificar si una semilla es de tipo Kama o no con las siguientes columnas: area, perimetro, compacto, longitud, coeficient.asimetria, longitud.ranura

Respuesta:

```
# Teniendo en cuenta la anterior división de los datos en train y test
# El modelo de RL
modelo <- glm(tipo ~ area + perimetro + compacto + longitud + coeficient.asimetria + longitud.ranura,
               data = df_seeds_train, family = binomial)

# Predicciones en el conjunto de prueba
predicciones <- predict(modelo, newdata = df_seeds_test, type = "response")

# Asignar valores de clase
predicciones_clase <- ifelse(predicciones > 0.5, "Kama", "No Kama")

# Resumen del modelo
summary(modelo)

Call:
glm(formula = tipo ~ area + perimetro + compacto + longitud +
    coeficient.asimetria + longitud.ranura, family = binomial,
    data = df_seeds_train)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) 671.4230   240.4966   2.792  0.00524 **
area         28.5841     7.4615   2.759  0.00580 **
perimetro    -32.0042    14.2003  -2.254  0.02421 *
compacto     -431.4114   157.0099  -2.748  0.00600 **
longitud     -59.7991    25.4630  -2.348  0.01885 *
coeficient.asimetria  1.8366    0.6386   2.876  0.00403 **
longitud.ranura  36.6699   14.0928   2.602  0.00927 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 184.239  on 146  degrees of freedom
Residual deviance: 23.053  on 140  degrees of freedom
AIC: 37.053

Number of Fisher Scoring iterations: 9
```

PREGUNTA 10 (1pt)

Si usamos un umbral de 0 en la salida del modelo (lo que equivale a probabilidad de 0.5 cuando usamos el predict con type='response') ¿Cuáles son los valores de precisión y exhaustividad?

Respuesta. Para calcular el accuracy y recall, primero necesitamos conocer los verdaderos positivos (TP), falsos positivos (FP) y falsos negativos (FN). Así que partiendo de la premisa de que "Kama" es la clase positiva y "no Kama" la negativa, tenemos:

$$\text{Precisión} = \text{TP}/(\text{TP}+\text{FP}) \text{ y } \text{Exhaustividad} = \text{TP}/(\text{TP}+\text{FN})$$

```
# Utilizar umbral de 0
predicciones_clase0 <- ifelse(predicciones > 0, "Kama", "No Kama")

# Cálculo TP, FP y FN
TP <- sum(predicciones_clase0 == "Kama" & df_seeds_test$tipo == "Kama")
FP <- sum(predicciones_clase0 == "Kama" & df_seeds_test$tipo == "No Kama")
FN <- sum(predicciones_clase0 == "No Kama" & df_seeds_test$tipo == "Kama")

# Cálculo precisión y exhaustividad
precision <- TP / (TP + FP)
exhaustividad <- TP / (TP + FN)

precision
```

```
[1] 1
```

```
exhaustividad
```

```
[1] 1
```

Seguramente estos resultados se deban a que estamos tratando con unos datos ultraperfectos, ya que es un dataset pequeño e ideado para hacer pruebas. Cuando nos enfrentamos a planteamientos reales en escenarios no tan perfectos, no suele alcanzarse el 1.

PREGUNTA 11 (1.5pt)

¿Qué están haciendo las siguientes líneas?

```
set.seed(123)
cl<-df_seeds |> select(area,perimetro,compacto,longitud,anchura,coeficient.asimetria,longitud.ra
table(real=df_seeds$tipo,cluster=cl$cluster)

cluster
real      1  2  3
  Kama     1 60  9
  Rosa    60 10  0
Canadian  0  2 68
```

Respuesta: Estamos usando clustering (agrupamiento), el algoritmo k-means. Primero establecemos una semilla aleatoria para asegurarnos de la reproducibilidad de resultados, ya que generamos resultados aleatorios donde es necesario. Luego hacemos clustering con k-means, con 3 clusters en las variables seleccionadas del conjunto de datos, es decir, que el algoritmo intentará agrupar las muestras en 3 grupos diferentes basándose en las características que le hemos indicado. Finalmente pasa a una tabla la relación entre los tipos de semillas y los clusters asignados por el algoritmo, para que podamos tener una visión de cómo se relacionan los grupos generados con los tipos de semillas en el conjunto de datos.

En la tabla de resultados podemos ver cuántas muestras de un tipo de semilla específico pertenecen a cada cluster (1, 2 o 3). En general, el modelo está agrupando bien las muestras, aunque hay un pelín de solapamiento.