

LABORATORIO DE DATOS

Primer Cuatrimestre 2024

Práctica N° 5: Modelo lineal multivariado. Entrenamiento y testeo.

1. Queremos estudiar la relación entre la longitud de la aleta de un pingüino y el peso del pingüino. Como en una esfera, el peso es proporcional a la longitud del radio elevada al cubo, podemos conjeturar que un polinomio de grado 3 es apropiado para ajustar el peso en función de la longitud de la aleta. Queremos verificar si nuestra conjetura tiene sustento en los datos.

- (a) **Datos faltantes.** Ejecutar el siguiente código y observar si hay filas con datos faltantes (NaN).

```
penguins = sns.load_dataset("penguins")
penguins.head()
```

Para hacerlo en forma más sistemática (en lugar de mirar solo algunas filas) puedes usar el siguiente código

```
penguins.isnull().values.any()
```

Para eliminar las filas con valores faltantes aplicamos al DataFrame el método `dropna()`. Eliminar las filas con datos faltantes del DataFrame de pingüinos y verificar que el DataFrame resultante no contiene valores faltantes.

- (b) Dividir el dataset resultante en un grupo de entrenamiento y uno de test (80% - 20%).
 - (c) Crear y ajustar 3 modelos utilizando polinomios de grados 1, 2 y 3.
 - (d) Calcular para cada uno el error predicción en el grupo de entrenamiento y en el grupo de test.
 - (e) ¿Cuál modelo tiene el menor error de ajuste? ¿Cuál el menor error de predicción?
 - (f) En base a los resultados obtenidos, ¿cuál de los tres modelos considera que es el más apropiado?
2. En el archivo `50_startups.csv` tenemos los siguientes datos de 50 compañías: gastos en investigación y desarrollo, gastos administrativos, gastos en marketing y ganancias. Queremos estimar las ganancias a partir de los gastos en las distintas áreas.

- (a) Leer el archivo, y realizar un gráfico de dispersión para cada par de variables. Se pueden generar todos los gráficos automáticamente con el `pairplot`.

```
sns.pairplot(
    data=vehicles, aspect = .8);
```

En base a estos gráficos, si quisiéramos predecir la ganancia mediante un modelo lineal utilizando una sola variable predictora, ¿cuál variable utilizaría? Diseñar un experimento para verificar su respuesta.

- (b) En este ejemplo, ¿considera que un modelo lineal multivariado ayudaría a predecir mejor la ganancia? Realizar un experimento para verificar su respuesta.
3. Se quiere predecir el peso de un pingüino usando como variables predictoras el largo de la aleta y el sexo del pingüino.

- (a) ¿Cuáles son todos los valores que toma la variable "sex"? ¿Qué tipo de variable es: numérica o categórica, ordinal o nominal? ¿Es una variable binaria?
- (b) Escribir (en lápiz y papel) la ecuación de un modelo lineal para este caso. ¿Qué unidades tienen las variables y cómo se codifica la variable "sexo del pingüino"?
- (c) **Codificación de variables binarias** Para crear una columna con el sexo codificado como 0 y 1, utilizamos el siguiente código:

```
from sklearn.preprocessing import OrdinalEncoder
encoder = OrdinalEncoder()
sex01 = encoder.fit_transform(penguins[["sex"]])
penguins["sex01"] = sex01
```

- (d) Ajustar el modelo usando todos los datos disponibles. Reportar los coeficientes encontrados y calcular el error de predicción (MSE).
- (e) Hacer un gráfico de los datos junto con las predicciones del modelo.
- (f) Dos pingüinos que tienen igual largo de aleta, uno macho y otro hembra, ¿qué diferencia de peso predice el modelo que tendrán?
4. Ahora se quiere predecir el peso de un pingüino usando como variables predictoras el largo de la aleta y la especie del pingüino.

- (a) Trabajamos con la base de pingüinos sin datos faltantes. ¿Cuáles son todos los valores que toma la variable "sex"? ¿Qué tipo de variable es: numérica o categórica, ordinal o nominal? ¿Es una variable binaria?
- (b) Escribir (en lápiz y papel) la ecuación de un modelo lineal para este caso. ¿Cómo se codifica la variable "especie"?
- (c) Explicar qué diferencia tiene este modelo respecto al propuesto en el ejercicio 1.
- (d) **Codificación de variables categóricas** Para agregar variables dummies para cada una de las especies usamos `OneHotEncoder`. Correr el siguiente código y verificar el resultado:

```
from sklearn.preprocessing import OneHotEncoder
penguins = sns.load_dataset("penguins").dropna()
encoderOHE = OneHotEncoder(sparse_output = False)
species3 = encoderOHE.fit_transform(penguins[["species"]])
5 species3_df = pd.DataFrame(species3,
    columns=encoderOHE.get_feature_names_out())
penguins3 = pd.concat([penguins, species3_df], axis = 1)
penguins3.head()
```

- (e) Ajustar el modelo usando todos los datos disponibles. Reportar los coeficientes encontrados y calcular el error de predicción.
- (f) Hacer un gráfico de los datos junto con las predicciones del modelo.