4.2.2.6 Práctica de laboratorio: evaluación de los errores de ajuste en la regresión lineal

Nevarez García Litzy Yulissa

Objetivos

En esta práctica de laboratorio, se familiarizará con los conceptos de evaluar los errores de ajuste en la regresión lineal.

- Parte 1: Importe las bibliotecas y los datos

- Parte 2: Calcule los errores

Aspectos básicos/situación

En las estadísticas, la regresión lineal es una manera de modelar una relación entre la variable dependiente y y la variable independiente x . El objetivo de la regresión es encontrar un modelo que describa los datos con la mayor precisión posible.

En esta práctica de laboratorio, utilizará los datos de ventas y el resultado de la regresión lineal de una práctica de laboratorio anterior para evaluar la precisión del modelo.

Recursos necesarios

- 1 computadora con acceso a Internet
- Bibliotecas de Python: pandas, numpy y sklearn
- Archivos de datos: stores-dist.csv

Parte 1: Importe las bibliotecas y los datos

En esta parte, importará las bibliotecas y los datos del archivo stores-dist.csv.

Paso 1: Importe las bibliotecas.

En este paso, importará las bibliotecas siguientes:

- numpy como np
- pandas como pd

In [1]:
```
# Code Cell 1
# This lab produces some minor warnings that can be ignored.
# These warnings appear because some libraries are updated more often than others
# and the system is letting the user know that some function will be depricated soon
```

```
# Use the following code to prevent the warnings from being displayed, or comment them
# to see the warnings
import warnings
warnings.filterwarnings('ignore')

# Import numpy and pandas
import numpy as np
import pandas as pd

from sklearn import cross_validation
from sklearn.linear_model import LinearRegression
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
Input In [1], in <cell line: 14>()
     11 import numpy as np
     12 import pandas as pd
---> 14 from sklearn import cross_validation
     15 from sklearn.linear_model import LinearRegression

ImportError: cannot import name 'cross_validation' from 'sklearn' (C:\ProgramData\Ana
conda3\lib\site-packages\sklearn\__init__.py)
```

Paso 2: Importe los datos.

En este paso, importará los datos del archivo stores-dist.csv, cambiará los encabezados de las columnas y verificará que el archivo se haya importado correctamente.

Los encabezados de las columnas ventas netas anuales y número de tiendas en el distrito se renombran para facilitar el procesamiento de los datos.

- ventas netas anuales a ventas
- número de tiendas en el distrito a tiendas

In [4]:
```
# Code Cell 2

# Import the file stores-dist.txt
salesDist = pd.read_csv('C:/Users/yulis/Analitica de los Datos en las organizaciones/s

# Change the column headings
salesDist.columns = ['district','sales','stores']

# Verify the imported data
salesDist.head()
```

Out[4]:

	district	sales	stores
0	1	231.0	12
1	2	156.0	13
2	3	10.0	16
3	4	519.0	2
4	5	437.0	6

La columna cdistrict no es necesaria para la evaluación del ajuste de la regresión lineal; por lo

tanto, esta columna puede descartarse.

In [5]:
```
# Code Cell 3
# Drop the district column.
sales = salesDist.drop('district',axis=1)

# Verify that the district column has been dropped.
sales.head()
```

Out[5]:

	sales	stores
0	231.0	12
1	156.0	13
2	10.0	16
3	519.0	2
4	437.0	6

Parte 2: Cálculo de errores

En esta parte, utilizará numpy para generar una línea de regresión para los datos analizados. También calculará el centroide para este conjunto de datos. El centroide es la media del conjunto de datos. La línea de regresión lineal simple generada también debe atravesar el centroide.

También utilizará sklearn.metrics para evaluar el modelo de regresión lineal. Calculará la puntuación R2 y el error medio cuadrático (MSE).

Paso 1: Asigne las variables x e y.

Asigne las ventas del marco de datos como la variable dependiente y y las tiendas del marco de datos como la variable independiente para el eje x .

In [7]:
```
# Code Cell 4
#dependent variable for y axis
y = sales.sales
#independent variable for x axis
x = sales.stores
```

Paso 2: Calcule los valores de y en el modelo

En la práctica de laboratorio anterior, calculó los componentes para el ajuste de la regresión lineal con un modelo polinomial mediante np.polyfit para calcular el vector de los coeficientes p que minimiza el error cuadrático. Mediante np.poly1d, puede calcular el valor correspondiente para cada valor de x en el modelo polinomial estimado.

Para recuperar la pendiente y la intersección y de la línea, utilice la variable p. La matriz p muestra el coeficiente en un orden descendente. Para un polinomio de primer orden, el primer coeficiente es la pendiente (m) y el segundo coeficiente es la intersección y (b).

In [8]:
```
# Code Cell 5
# compute the y values from the polynomial model for each x value
order = 1
p = np.poly1d(np.polyfit(x, y ,order))

print('The array p(x) stores the calculated y value from the polynomial model for each
print('\nThe vector of coefficients p describes this regression model:\n{}'.format(p))
print('\nThe zeroth order term (y-intercept or b) is stored in p[0]: {}.'.format(p[0]))
print('\nThe first order term (slope or m) is stored in p[1]: {}.'.format(p[1]))
```

```
The array p(x) stores the calculated y value from the polynomial model for each x val
ue,

[169.93468442 134.14759895  26.78634257 527.80553905 384.65719719
 420.44428266 205.72176988 134.14759895  26.78634257 277.29594081
 527.80553905 313.08302627 456.23136812  62.57342803 169.93468442
 205.72176988 420.44428266  98.36051349 313.08302627 527.80553905
 563.59262451  62.57342803 134.14759895 348.87011173 384.65719719
 563.59262451 277.29594081].

The vector of coefficients p describes this regression model:

-35.79 x + 599.4

The zeroth order term (y-intercept or b) is stored in p[0]: 599.3797099726614.

The first order term (slope or m) is stored in p[1]: -35.787085462974005.
```

Paso 3: Utilice diversas medidas para evaluar los modelos.

En este paso, utilizará sklearn para evaluar los modelos. Sklean ofrece una variedad de medidas. Calculará la puntuación R2 , el error medio cuadrático (MSE) y el error medio absoluto (MAE) con las funciones en sklearn.

Para calcular el valor para cada medida, indique los valores de y, que son los valores obtenidos del archivo csv importado, stores-dist.csv como el primer argumento. Como segundo argumento, utilice los valores de p(x), que se calcularon de su modelo polinomial de primer orden en la siguiente forma:

y=mx+b

donde m es p[1] y b es p[0] en los resultados de poly1d.

La función de puntuación de regresión R2 (coeficiente de determinación) ofrece cierta información sobre la cantidad de ajuste del modelo. La mejor puntuación para R2 es 1,0. Esta puntuación indica qué tan bien explica el modelo el resultado obtenido.

In [9]:
```
# Code Cell 6
from sklearn.metrics import r2_score
r2 = r2_score(y, p(x))
r2
```

Out[9]:
```
0.83217523508888
```

El error medio cuadrático (MSE) es una medida de qué tan bien se puede usar el modelo para realizar una predicción. Este número siempre es no negativo. Los mejores valores se encuentran más cercanos a cero.

In [10]:
```
# Code Cell 7
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y, p(x))
mse
```

Out[10]:
```
5961.386465941158
```

El error medio absoluto (MAE) es una medida de cuánto se acercan las predicciones a los resultados eventuales. El MAE es un promedio de los errores absolutos entre la predicción y el verdadero valor.

In [11]:
```
# Code Cell 8
from sklearn.metrics import mean_absolute_error
mae = mean_absolute_error(y, p(x))
mae
```

Out[11]:
```
61.2232611786873
```

Todas estas medidas permiten que determine qué tan bien puede su modelo hacer la predicción. En esta práctica de laboratorio, se evaluó sólo un modelo, regresión lineal simple.