Julian Peña Reyes

Análisis de datos

TALLER 4

Elaborar en Python la regresión múltiple para el ejercicio propuesto en clase.

VAR RESPUESTA		
cantidad vendida	price	advertiseing
8500	$2.00	2800
4700	$5.00	200
5800	$3.00	400
7400	$2.00	500
6200	$5.00	3200
7300	$3.00	1800
5600	$4.00	900

Solución

Creamos nuestro código para agregar los datos en 3 columnas y crear el modelo de regresión multivariada.

```
import pandas as pd
import numpy as np
import sklearn.linear_model as LinearRegression
import matplotlib.pyplot as plt
from sklearn import datasets, linear_model
import statsmodels.api as sm
import statsmodels.stats.diagnostic as smd

Q =  np.array([8500,4700,5800,7400,6200,7300,5600])
P = np.array([2,5,3,2,5,3,4])
A = np.array([2800,200,400,500,3200,1800,900])

X_multiple = pd.DataFrame({"P":P,"A": A})
```

```
print(X_multiple.describe())


y_multiple = Q


from sklearn.model_selection import train_test_split

#Separo los datos de "train" en entrenamiento y prueba para probar los
algoritmos

X_train, X_test, y_train, y_test = train_test_split(X_multiple, y_multiple,
test_size=0.2)


#Defino el algoritmo a utilizar

lr_multiple = linear_model.LinearRegression()


#Entreno el modelo

lr_multiple.fit(X_train, y_train)



#Realizo una predicción

Y_pred_multiple = lr_multiple.predict(X_test)


print('DATOS DEL MODELO REGRESIÓN LINEAL MULTIPLE')

print()

print('Valor de las pendientes o coeficientes "a":')

print(lr_multiple.coef_)

print('Valor de la intersección o coeficiente "b":')

print(lr_multiple.intercept_)


print('Precisión del modelo:')

print(lr_multiple.score(X_train, y_train))


X_train = sm.add_constant(X_train, prepend=True)

modelo = sm.OLS(endog=y_train, exog=X_train,)

modelo = modelo.fit()
```

```
print(modelo.summary())
```

Resultados

```
              P            A
count  7.000000     7.000000
mean   3.428571  1400.000000
std    1.272418  1215.181742
min    2.000000   200.000000
25%    2.500000   450.000000
50%    3.000000   900.000000
75%    4.500000  2300.000000
max    5.000000  3200.000000
DATOS DEL MODELO REGRESIÓN LINEAL MULTIPLE

Valor de las pendientes o coeficientes "a":
[-484.36271108    0.94957206]
Valor de la intersección o coeficiente "b":
6868.355308813325
Precisión del modelo:
0.9920185674341906
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.992
Model:                            OLS   Adj. R-squared:                  0.984
Method:                 Least Squares   F-statistic:                     124.3
Date:                Thu, 03 Dec 2020   Prob (F-statistic):            0.00798
Time:                        20:37:00   Log-Likelihood:                -31.056
No. Observations:                   5   AIC:                             68.11
Df Residuals:                       2   BIC:                             66.94
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
```

const	6868.3553	634.902	10.818	0.008	4136.591	9600.120
P	-484.3627	139.786	-3.465	0.074	-1085.812	117.086
A	0.9496	0.148	6.422	0.023	0.313	1.586

Omnibus:	nan	Durbin-Watson:	2.669
Prob(Omnibus):	nan	Jarque-Bera (JB):	0.199
Skew:	-0.105	Prob(JB):	0.905
Kurtosis:	2.045	Cond. No.	1.18e+04

Si lo comparamos con la regresión hecha en Excel, podemos ver algunos cambios en los coeficientes y en el valor de la intersección que son en su mayoría debido a que en Python se entrena y se ajusta el modelo `lr_multiple.fit(X_train, y_train)` y esto se demuestra en que la precisión del modelo en Python es de 0.9920185674341906 y en Excel de 0.961736068.

Así mismo al hacer el análisis de residuales se ve la diferencia en grados de libertad y suma de cuadrados.