modelo-estadistico

September 2, 2023

1 Construcción de un modelo estadístico base

1. De acuerdo con la pregunta base, contempla la herramienta estadística necesaria para contestarla.
2. Selecciona al menos dos de las herramientas estadísticas que hemos analizado en el curso: regresión lineal simple y múltiple, anova o pruebas de hipótesis (medias o proporción). Justifica la elección de la herramienta estadística.
3. Valida el modelo obtenido analizando los supuestos requeridos por el modelo.
4. Haz uso de toda la herramienta estadística que creas necesaria. En caso de que requieras herramienta estadística no contemplada aún en el curso, consulta con tu profesora.
5. Grafica tus resultados para una mejor visualización de tus resultados
6. Interpreta en el contexto del problema toda la herramienta estadística que uses (no sólo coloques gráficos y tablas: indica cómo te ayudan al análisis)
7. Emite una conclusión del análisis que realizaste

NOTA IMPORTANTE: Perdón maestra, no pude avanzar más porque no supe qué herramienta de las que vimos utilizar. También creo que no hice bien la selección de variables debido a que nada más las seleccioné en base a un criterio, la correlación. En este fin de semana voy a tratar de corregir la primera evidencia y terminar esta. Muchas gracias por entender, que tenga bonito día.

1.1 Importar módulos

[1]:
```
# Importación de librerías
import pandas as pd
import numpy as np
import random as rnd
import math
import seaborn as sns
import statsmodels.api as sm
import scipy.stats as stats
import matplotlib.pyplot as plt
```

1.2 Cargar datos

[2]:
```
# Este bloque de codigo no es necesario si el archivo esta guardado localmente
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[3]:
```
# Dataframe del conjunto de datos
autos_df = pd.read_csv('/content/drive/MyDrive/TC3006C101_A00828096/Estadistica/
 ↪precios_autos.csv')

selected_columns = autos_df[['enginesize', 'curbweight', 'horsepower',␣
 ↪'carwidth', 'highwaympg', 'citympg', 'price']]
selected_columns
```

[3]:

	enginesize	curbweight	horsepower	carwidth	highwaympg	citympg	price
0	130	2548	111	64.1	27	21	13495.0
1	130	2548	111	64.1	27	21	16500.0
2	152	2823	154	65.5	26	19	16500.0
3	109	2337	102	66.2	30	24	13950.0
4	136	2824	115	66.4	22	18	17450.0
..	…	…	…	…	…	…	…
200	141	2952	114	68.9	28	23	16845.0
201	141	3049	160	68.8	25	19	19045.0
202	173	3012	134	68.9	23	18	21485.0
203	145	3217	106	68.9	27	26	22470.0
204	141	3062	114	68.9	25	19	22625.0

[205 rows x 7 columns]

1.3 Análisis de datos y Pregunta base

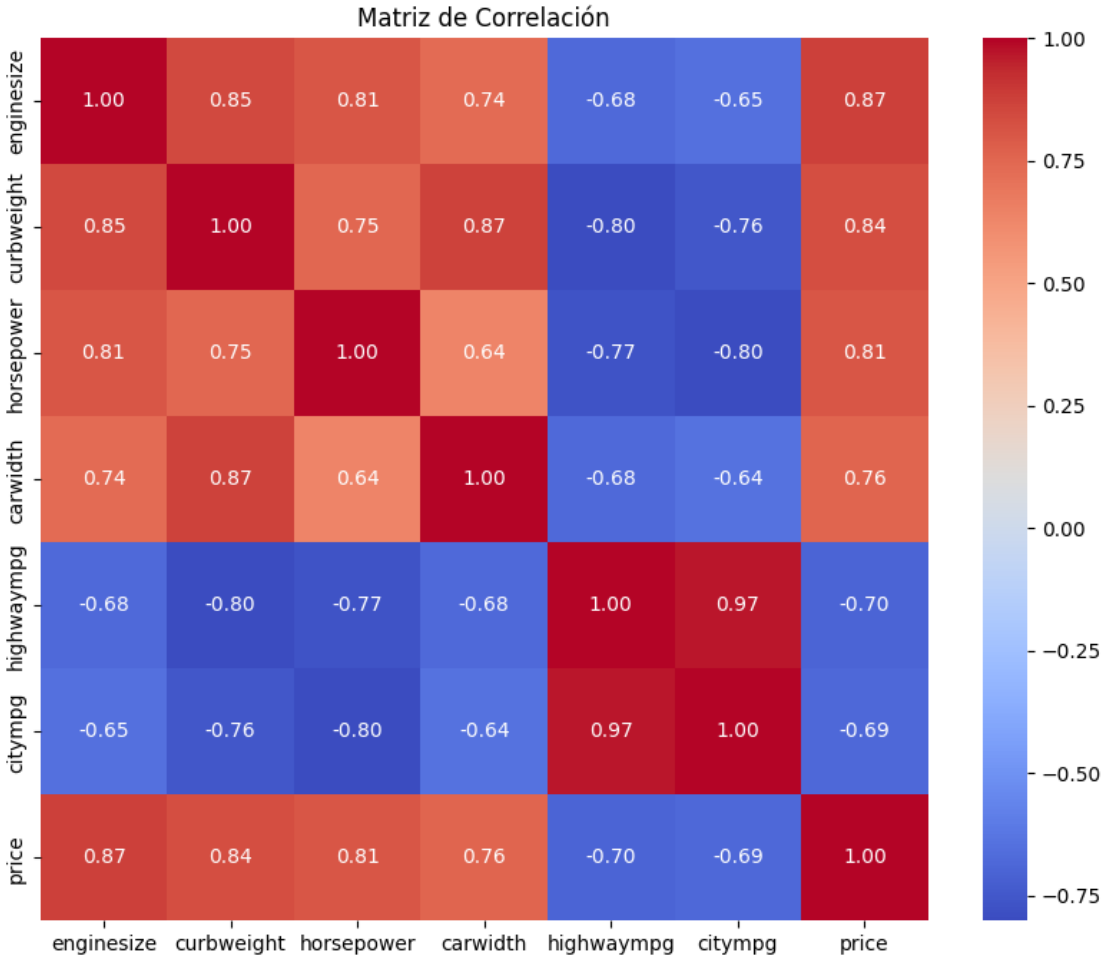
1.3.1 1. De acuerdo a la pregunta base, contempla la herramienta estadística necesaria para contestarla.

1.3.2 2. Selecciona al menos dos de las herramientas estadísticas que hemos analizado en el curso: regresión lineal simple y múltiple, anova o pruebas de hipótesis (medias o proporción). Justifica la elección de la herramienta estadística.

Condiciones para la regresión lineal múltiple

[4]:
```
correlation_matrix = autos_df[['enginesize', 'curbweight', 'horsepower',␣
 ↪'carwidth', 'highwaympg', 'citympg', 'price']].corr()

plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Matriz de Correlación')
plt.show()
```



Modelo de regresión lineal múltiple

1.3.3 3. Valida el modelo obtenido analizando los supuestos requeridos por el modelo.

1.3.4 4. Conclusión del análisis

[5]:
```
# Define the formula for linear regression
formula = "autos_df['price'] ~ autos_df['enginesize'] + autos_df['wheelbase'] +␣
 ↪autos_df['peakrpm']"

# Fit the linear regression model
model = sm.OLS.from_formula(formula, data=autos_df)
result = model.fit()
print(result.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:      autos_df['price']   R-squared:                       0.801
Model:                            OLS   Adj. R-squared:                  0.798
Method:                 Least Squares   F-statistic:                     270.1
Date:                Sat, 02 Sep 2023   Prob (F-statistic):           3.06e-70
Time:                        05:16:54   Log-Likelihood:                -1966.9
No. Observations:                 205   AIC:                             3942.
Df Residuals:                     201   BIC:                             3955.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
==========
                             coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
----------
Intercept              -4.512e+04   6326.572     -7.131      0.000   -5.76e+04
-3.26e+04
autos_df['enginesize']   156.7447      7.348     21.333      0.000     142.257
171.233
autos_df['wheelbase']    234.7686     52.817      4.445      0.000     130.622
338.916
autos_df['peakrpm']        2.9883      0.565      5.286      0.000       1.873
4.103
==============================================================================
Omnibus:                       23.280   Durbin-Watson:                   0.827
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               34.414
Skew:                           0.675   Prob(JB):                     3.37e-08
Kurtosis:                       4.485   Cond. No.                     1.30e+05
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

[2] The condition number is large, 1.3e+05. This might indicate that there are strong multicollinearity or other numerical problems.

[7]:
```
formula2 = "autos_df['price'] ~ autos_df['enginesize'] + autos_df['wheelbase']"

# Fit the linear regression model
model2 = sm.OLS.from_formula(formula2, data=autos_df)
result2 = model2.fit()
print(result2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:      autos_df['price']   R-squared:                       0.774
Model:                            OLS   Adj. R-squared:                  0.771
Method:                 Least Squares   F-statistic:                     345.2
Date:                Sat, 02 Sep 2023   Prob (F-statistic):           6.86e-66
Time:                        05:19:02   Log-Likelihood:                -1980.2
No. Observations:                 205   AIC:                             3966.
Df Residuals:                     202   BIC:                             3976.
Df Model:                           2
Covariance Type:            nonrobust
================================================================================
==========
                               coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
----------
Intercept                -2.19e+04   4846.976     -4.518      0.000   -3.15e+04
-1.23e+04
autos_df['enginesize']   154.7476      7.812     19.810      0.000     139.345
170.151
autos_df['wheelbase']    157.3072     54.021      2.912      0.004      50.790
263.824
==============================================================================
Omnibus:                       38.962   Durbin-Watson:                   0.777
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               70.509
Skew:                           0.963   Prob(JB):                     4.89e-16
Kurtosis:                       5.132   Cond. No.                     2.99e+03
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 2.99e+03. This might indicate that there are
strong multicollinearity or other numerical problems.
```

[9]:
```
formula3 = "autos_df['price'] ~ autos_df['enginesize']"

# Fit the linear regression model
model3 = sm.OLS.from_formula(formula3, data=autos_df)
result3 = model3.fit()
print(result3.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:      autos_df['price']   R-squared:                       0.764
Model:                            OLS   Adj. R-squared:                  0.763
Method:                 Least Squares   F-statistic:                     657.6
Date:                Sat, 02 Sep 2023   Prob (F-statistic):           1.35e-65
Time:                        05:22:09   Log-Likelihood:                -1984.4
No. Observations:                 205   AIC:                             3973.
Df Residuals:                     203   BIC:                             3979.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
==========
                             coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
----------
Intercept              -8005.4455    873.221     -9.168      0.000   -9727.191
-6283.700
autos_df['enginesize']   167.6984      6.539     25.645      0.000     154.805
180.592
==============================================================================
Omnibus:                       23.788   Durbin-Watson:                   0.768
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               33.092
Skew:                           0.717   Prob(JB):                     6.52e-08
Kurtosis:                       4.348   Cond. No.                         429.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```