# MODELOS DE REGRESIÓN LINEAL MULTIPLE

GRUPO 6

# AGENDA

Preparación del modelo

Modelo 1

Modelo 2

Modelo 3

Matríz de correlación

Comparación de los 3 modelos

# LIBRERIAS

```
#importar librerias
import numpy as np
import pandas as pd
import geopandas as gpd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import missingno as msno
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
import statsmodels.api as sm
```

# PREPARACIÓN DEL MODELO

In [246]:
```
df_caba = df_caba.loc[:, ['surface_total_in_m2', 'surface_covered_in_m2', 'price','lat','lon', 'rooms']]
```

In [247]:
```
df_caba.head(10)
```

Out[247]:

| | surface_total_in_m2 | surface_covered_in_m2 | price | lat | lon | rooms |
|---|---|---|---|---|---|---|
| **13** | 50.0 | 30.0 | 111700.0 | NaN | NaN | 1 room |
| **14** | 42.0 | 31.0 | 147900.0 | NaN | NaN | 1 room |
| **19** | 104.0 | 96.0 | 350000.0 | -34.580504 | -58.405874 | 3 rooms |
| **21** | 118.0 | 73.0 | 270500.0 | -34.590926 | -58.411665 | +4 rooms |
| **241** | 39.0 | 35.0 | 147300.0 | -34.588862 | -58.412307 | 1 room |
| **256** | 175.0 | 175.0 | 440000.0 | -34.566479 | -58.434075 | +4 rooms |
| **266** | 47.0 | 41.0 | 135000.0 | -34.576504 | -58.431468 | S/I |
| **282** | NaN | 153.0 | 770000.0 | NaN | NaN | +4 rooms |
| **386** | 70.0 | 61.0 | 179000.0 | -34.590243 | -58.436402 | 3 rooms |
| **443** | NaN | 175.0 | 379900.0 | -34.584641 | -58.411582 | +4 rooms |

```
df_caba.dropna(inplace=True)
```

```
df_caba = pd.get_dummies(df_caba, drop_first=True)
```

```
df_caba.head()
```

| | surface_total_in_m2 | surface_covered_in_m2 | price | lat | lon | rooms_1 room | rooms_2 room | rooms_3 rooms | rooms_S/I |
|---|---|---|---|---|---|---|---|---|---|
| **19** | 104.0 | 96.0 | 350000.0 | -34.580504 | -58.405874 | 0 | 0 | 1 | 0 |
| **21** | 118.0 | 73.0 | 270500.0 | -34.590926 | -58.411665 | 0 | 0 | 0 | 0 |
| **241** | 39.0 | 35.0 | 147300.0 | -34.588862 | -58.412307 | 1 | 0 | 0 | 0 |
| **256** | 175.0 | 175.0 | 440000.0 | -34.566479 | -58.434075 | 0 | 0 | 0 | 0 |
| **266** | 47.0 | 41.0 | 135000.0 | -34.576504 | -58.431468 | 0 | 0 | 0 | 1 |

```
df_caba.to_csv('df_caba.csv')
```

# EVALUACIÓN MODELO 1

```
1
2  X_1 = df_caba.drop(columns = ['price'])
3  y_1 = df_caba['price']
4
5
6  Xtrain_1, Xtest_1, ytrain_1, ytest_1 = train_test_split(X_1, y_1, random_state=123)
7  scaler = StandardScaler()
8
9  Xtrain_1 = scaler.fit_transform(Xtrain_1)
10 Xtest_1 = scaler.transform(Xtest_1)
11
12
13 # Tenemos que agregar explícitamente a una constante:
14 Xtrain_1 = sm.add_constant(Xtrain_1)
15 Xtest_1 = sm.add_constant(Xtest_1)
```

```
1  # Entrenamos el modelo 1
2
3  model_1 = sm.OLS(ytrain_1, Xtrain_1).fit()
4
5  display(model_1.summary())
```

OLS Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | price | **R-squared:** | 0.286 |
| **Model:** | OLS | **Adj. R-squared:** | 0.281 |

# EVALUACIÓN MODELO 2 CON INTERACIÓN

```
X_2 = df_caba.drop(columns = ['price',])
X_2['surface_total_in_m2 * surface_covered_in_m2'] = X_2.surface_total_in_m2 * X_2.surface_covered_in_m2
y_2 = df_caba['price']


Xtrain_2, Xtest_2, ytrain_2, ytest_2 = train_test_split(X_2, y_2, random_state=123)
scaler = StandardScaler()

Xtrain_2 = scaler.fit_transform(Xtrain_2)
Xtest_2 = scaler.transform(Xtest_2)


# Tenemos que agregar explícitamente a una constante:
Xtrain_2 = sm.add_constant(Xtrain_2)
Xtest_2 = sm.add_constant(Xtest_2)
```

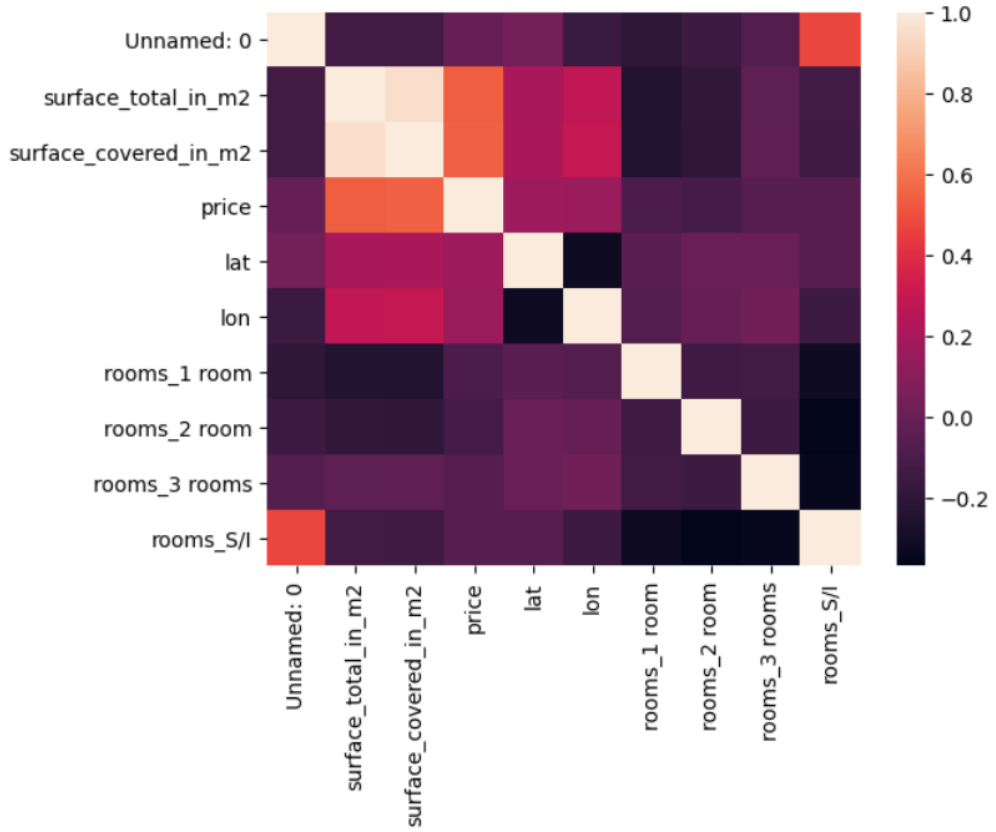
```
# Entrenamos el modelo 2 (con interaccion)

model_2 = sm.OLS(ytrain_2, Xtrain_2).fit()

display(model_2.summary())
```

OLS Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | price | **R-squared:** | 0.287 |
| **Model:** | OLS | **Adj. R-squared:** | 0.282 |

# MATRÍZ DE CORRELACIÓN

```
Unnamed: 0                -0.01
surface_total_in_m2        0.54
surface_covered_in_m2      0.54
price                      1.00
lat                        0.17
lon                        0.16
rooms_1 room              -0.09
rooms_2 room              -0.12
rooms_3 rooms             -0.06
rooms_S/I                 -0.06
Name: price, dtype: float64
```



# EVALUACIÓN MODELO 3

```python
X_3 = df_caba[['surface_total_in_m2', 'surface_covered_in_m2' ]]
y_3 = df_caba['price']


Xtrain_3, Xtest_3, ytrain_3, ytest_3 = train_test_split(X_3, y_3, random_state=123)
scaler = StandardScaler()

Xtrain_3 = scaler.fit_transform(Xtrain_3)
Xtest_3 = scaler.transform(Xtest_3)


# Tenemos que agregar explícitamente a una constante:
Xtrain_3 = sm.add_constant(Xtrain_3)
Xtest_3 = sm.add_constant(Xtest_3)
```

```python
# Entrenamos el modelo 2 (con interaccion)

model_3 = sm.OLS(ytrain_3, Xtrain_3).fit()

display(model_3.summary())
```

OLS Regression Results

| | | | |
|---|---|---|---|
| **Dep. Variable:** | price | **R-squared:** | 0.273 |
| **Model:** | OLS | **Adj. R-squared:** | 0.272 |

# COMPARACIÓN

```
ypred_1 = model_1.predict(Xtest_1)

# Error cuadrático medio modelo 1
print("Error cuadrático medio modelo 1: %.2f" % mean_squared_error(ytest_1, ypred_1))
# Evaluamos el puntaje de varianza (siendo 1.0 el mejor posible)
print('Puntaje de varianza modelo 1: %.2f' % r2_score(ytest_1, ypred_1))

```

```
Error cuadrático medio modelo 1: 94333630979.22
Puntaje de varianza modelo 1: 0.52
```

```
ypred_2 = model_2.predict(Xtest_2)

# Error cuadrático medio modelo 2
print("Error cuadrático medio modelo 2: %.2f" % mean_squared_error(ytest_2, ypred_2))
# Evaluamos el puntaje de varianza (siendo 1.0 el mejor posible)
print('Puntaje de varianza modelo 2: %.2f' % r2_score(ytest_2, ypred_2))
```

```
Error cuadrático medio modelo 2: 92194302157.35
Puntaje de varianza modelo 2: 0.53
```

```
ypred_3 = model_3.predict(Xtest_3)

# Error cuadrático medio modelo 3
print("Error cuadrático medio modelo 3: %.2f" % mean_squared_error(ytest_3, ypred_3))
# Evaluamos el puntaje de varianza (siendo 1.0 el mejor posible)
print('Puntaje de varianza modelo 3: %.2f' % r2_score(ytest_3, ypred_3))
```

```
Error cuadrático medio modelo 3: 93073271174.76
Puntaje de varianza modelo 3: 0.52
```