# Apartment for Rent Classified

El siguiente informe describe el proceso realizado, para hallar una modelo predictora, para el precio de renta de apartamentos y casas.

Los datos fueron obtenidos de:

https://archive.ics.uci.edu/dataset/555/apartment+for+rent+classified

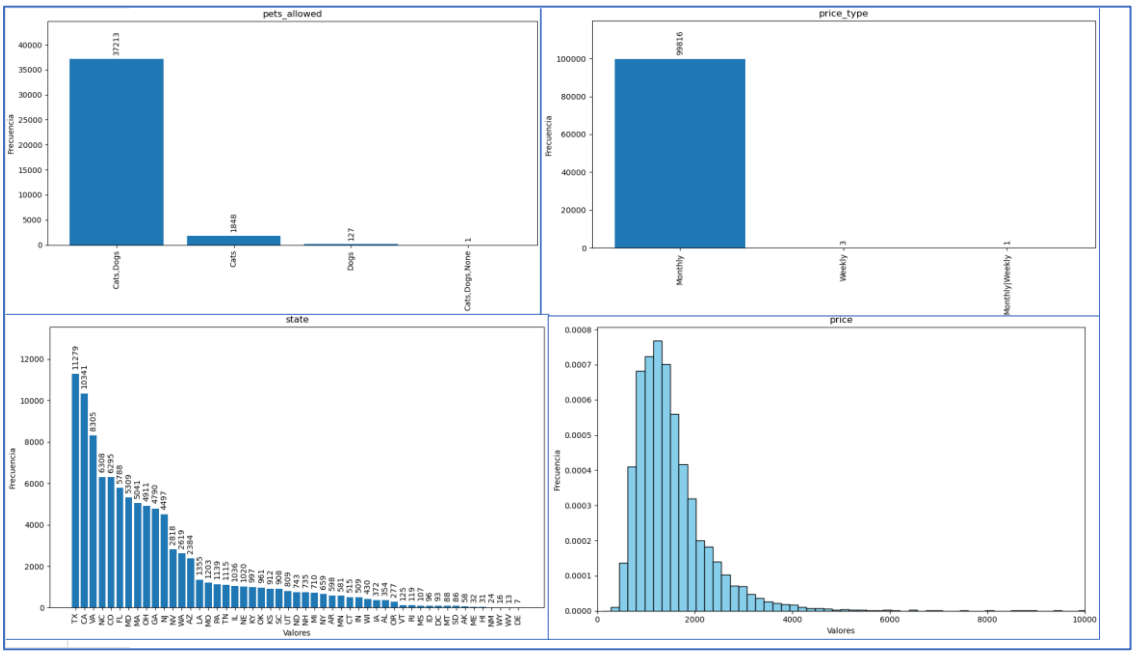
El set de datos cuenta con 99.820 registros y 22 atributos, los cuales son:

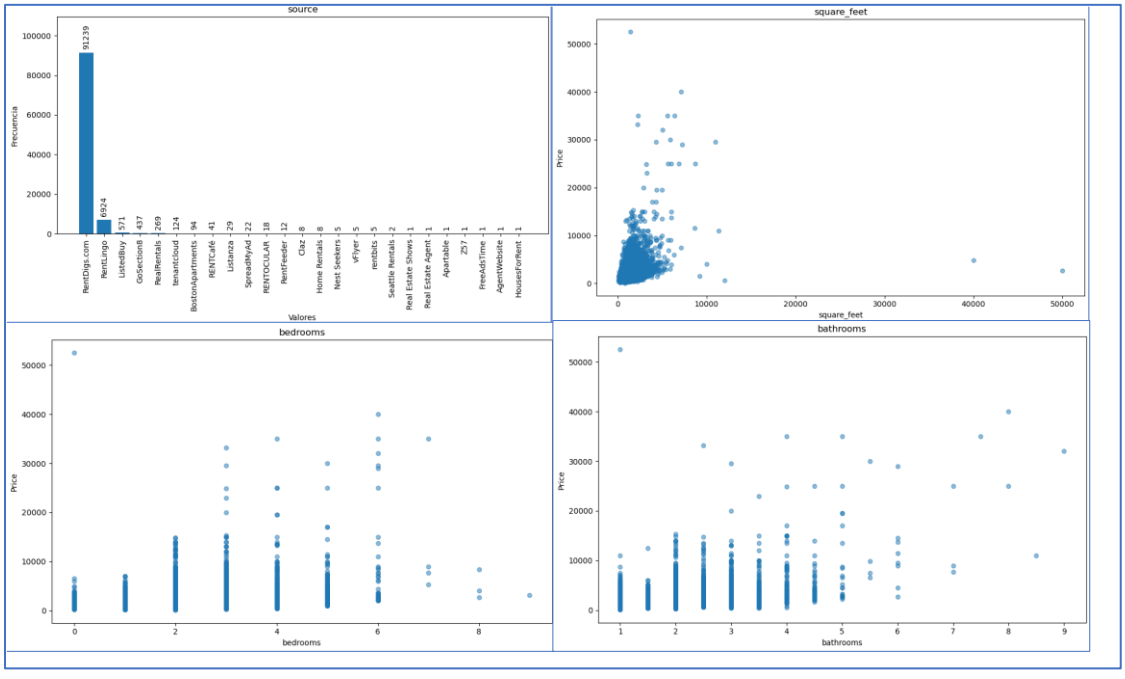
| | |
|---|---|
| id | price |
| category | price_display |
| title | price_type |
| body | square_feet |
| amenities | address |
| bathrooms | cityname |
| bedrooms | state |
| currency | latitude |
| fee | longitude |
| has_photo | source |
| pets_allowed | time |

**exploración.py**

Al ejecutar el script, se obtienen entre otros los siguientes datos.

```
La cantidad de valores unicos por cada atributos es:
id: 99736
category: 7
title: 58656
body: 94801
amenities: 9841
bathrooms: 16
bedrooms: 10
currency: 1
fee: 2
has_photo: 3
pets_allowed: 4
price: 3691
price_display: 1511
price_type: 3
square_feet: 2539
address: 2775
cityname: 2981
state: 51
latitude: 7213
longitude: 7272
source: 25
time: 75541

Resumen para el atributo: bathrooms
count    99757.000000
mean         1.445332
std          0.546986
min          1.000000
25%          1.000000
50%          1.000000
75%          2.000000
max          9.000000
Name: bathrooms, dtype: float64

Resumen para el atributo: bedrooms
count    99696.000000
mean         1.727983
std          0.748941
min          0.000000
25%          1.000000
50%          2.000000
75%          2.000000
max          9.000000
Name: bedrooms, dtype: float64

Resumen para el atributo: bedrooms
count    99696.000000
mean         1.727983
std          0.748941
min          0.000000
25%          1.000000
50%          2.000000
75%          2.000000
max          9.000000
Name: bedrooms, dtype: float64

Resumen para el atributo: price
count    99819.000000
mean      1527.224015
std        903.638141
min        100.000000
25%       1014.000000
50%       1350.000000
75%       1795.000000
max      52500.000000
Name: price, dtype: float64
```



**preprocesado.py**

para este script, lo que se hace es eliminar atributos que no se consideran que puedan ser relevantes para calcular los precios, o registros que contenían información incompleta o que podrían ser problemáticos en las siguientes etapas de aprendizaje.

```
#Eliminacion de atributtos irrelevantes.
L_eliminar = ['id',
              'title',
              'body',
              'amenities',
              'currency',
              'price_display',
              'address',
              'cityname',
              'latitude',
              'longitude',
              'time']
dataset = dataset.drop( columns = L_eliminar)

#Eliminar registro con el precio mas alto.
precioMaximo = dataset['price'].max()
indicePrecioMaximo = dataset[dataset['price'] == precioMaximo].index
dataset = dataset.drop(indicePrecioMaximo)

# Eliminar los registros donde el atributo 'category' es 'housing/rent/apartment'
dataset = dataset[dataset['category'] == 'housing/rent/apartment']

# Eliminar los registros donde el atributo 'bedrooms' es 7 o más
dataset = dataset[dataset['bedrooms'] < 7]

# Eliminar registros con valores faltantes
dataset = dataset.dropna()

# Eliminar registros con valores WV
dataset = dataset[dataset['state'] != 'WV']
```

**modelo1.py**

En este, se crea un algoritmo de regresión **lineal con las variables**, independientes.

```
X = dataset[['state', 'bathrooms', 'bedrooms', 'square_feet']]
```

Los resultados, para el modelo son los siguientes.

```
Error cuadrático medio: 239479.9375994268
Coeficiente de determinación (R^2): 0.4982100431904015

In [353]:
```

**Modelo2.py**

En este, se crea un algoritmo de random forest, para calcular el precio, también con las variables.

```
X = dataset[['state', 'bathrooms', 'bedrooms', 'square_feet']]
```

```
Mejores hiperparámetros: <bound method BaseEstimator.get_params of RandomizedSearchCV(cv=3,
estimator=RandomForestRegressor(random_state=42),
                   n_iter=100, n_jobs=-1,
                   param_distributions={'bootstrap': [True, False],
                                        'max_depth': [None, 10, 20, 30, 40, 50],
                                        'max_features': ['auto', 'sqrt',
                                                         'log2'],
                                        'min_samples_leaf': [1, 2, 4],
                                        'min_samples_split': [2, 5, 10],
                                        'n_estimators': [100, 200, 300, 400,
                                                         500]},
                   random_state=42, verbose=2)>
Error cuadrático medio (Random Forest): 212765.74773718455
Coeficiente de determinación (R^2) (Random Forest): 0.5541851378540725

In [354]:
```

**Conclusion**

Como se puede observar, ambos modelos presentaron un r^2 no tan cercano a uno, por lo que se puede decir que los modelos aun podrían tener margen de mejora.

0.55 para random forest

0.49 para regresión lineal.