Trabajo Final Henry Machine Learning

En este documento se explicara detalladamente los pasos realizados en le modelo y el porque de cada uno.

- El primer paso fue importar el modelo el cual tuve que importar con una codificacion un tanto extraña ya que de no hacerlo de esta forma se importaban los datos de precios con la notacion cientifica, no se importaban los datos en valores numericos sino que se importaban por ejemplo:

  Precio: 300045e+442.

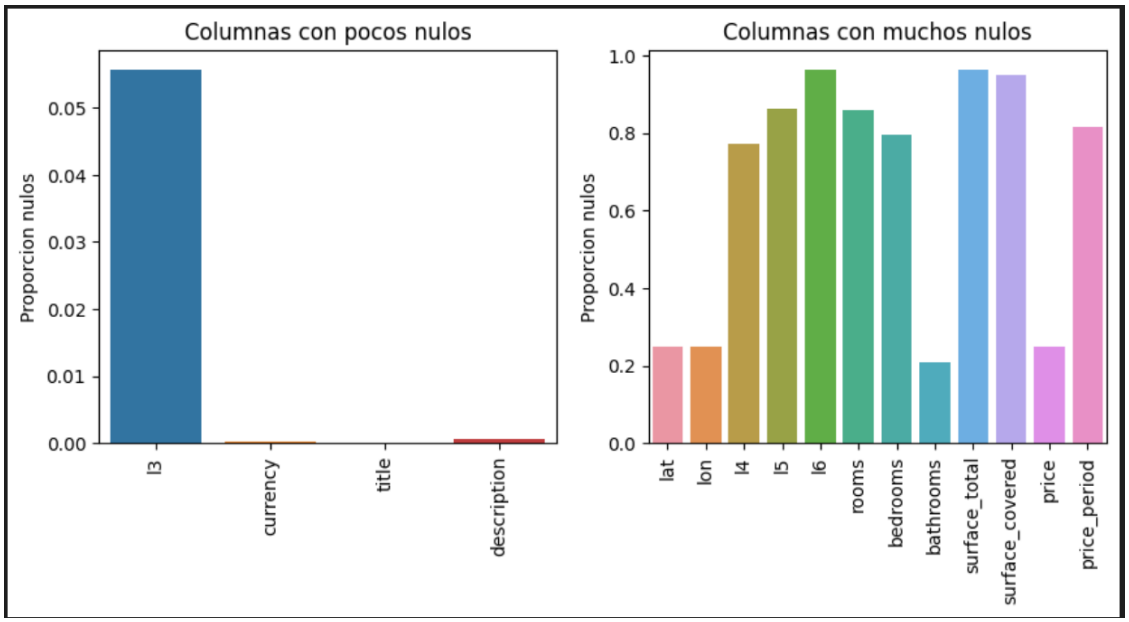
  Ese problema se soluciono de manera muy sencilla con el siguiente comando:

```
pd.set_option('display.float_format','{:2f}'.format)
```

- Luego realice una concatenacion de las dos tablas las cuales se llamaban properties_colombia_test.csv y properties_colombia_train.csv

```
data_train_test = pd.concat([train_data, test_data],axis=0,ignore_index=True)
```

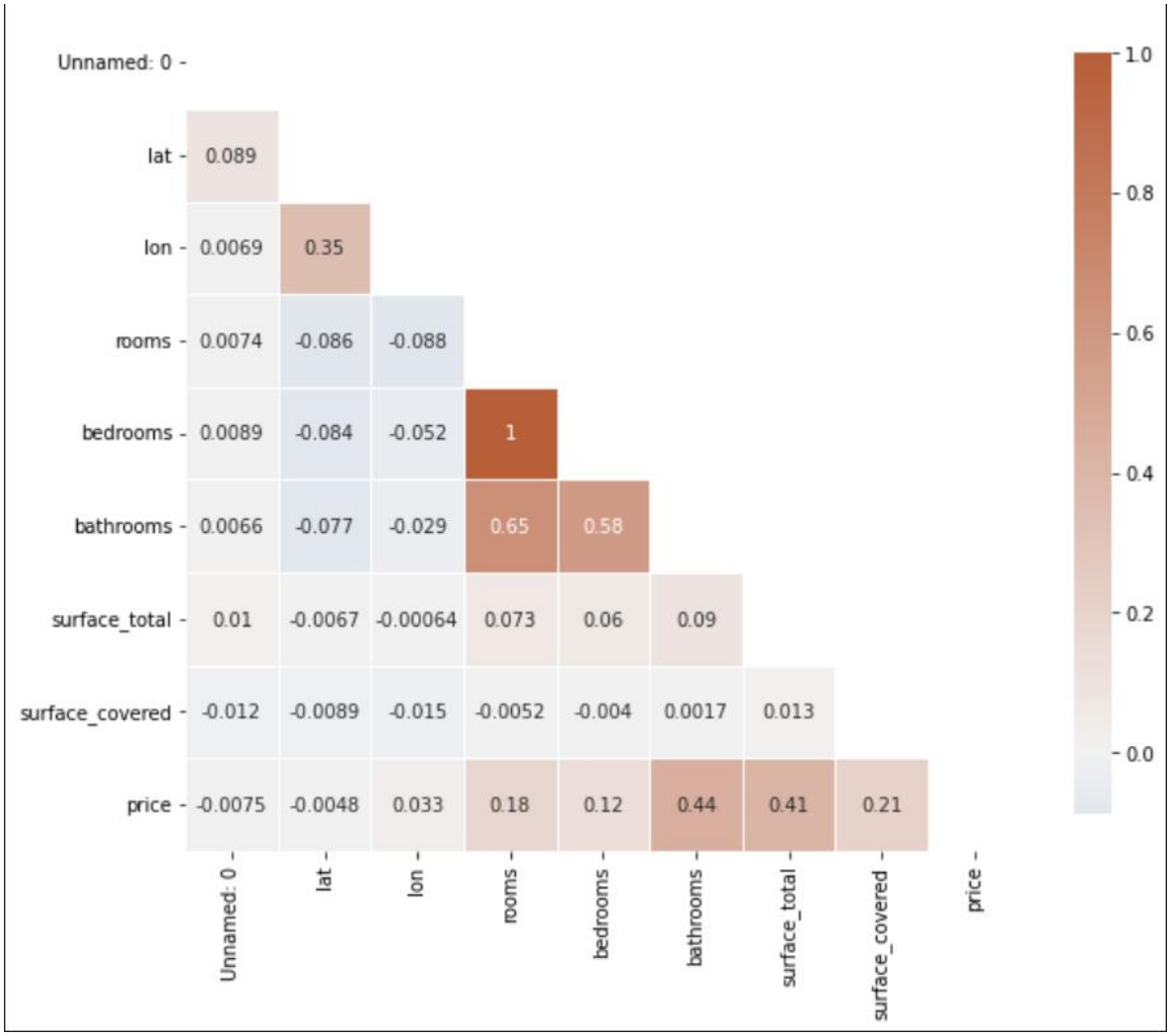
- Luego de eso hice una grafica para evaluar la cantidad de nulos en cada columna de la tabla concatenada anteriormente:

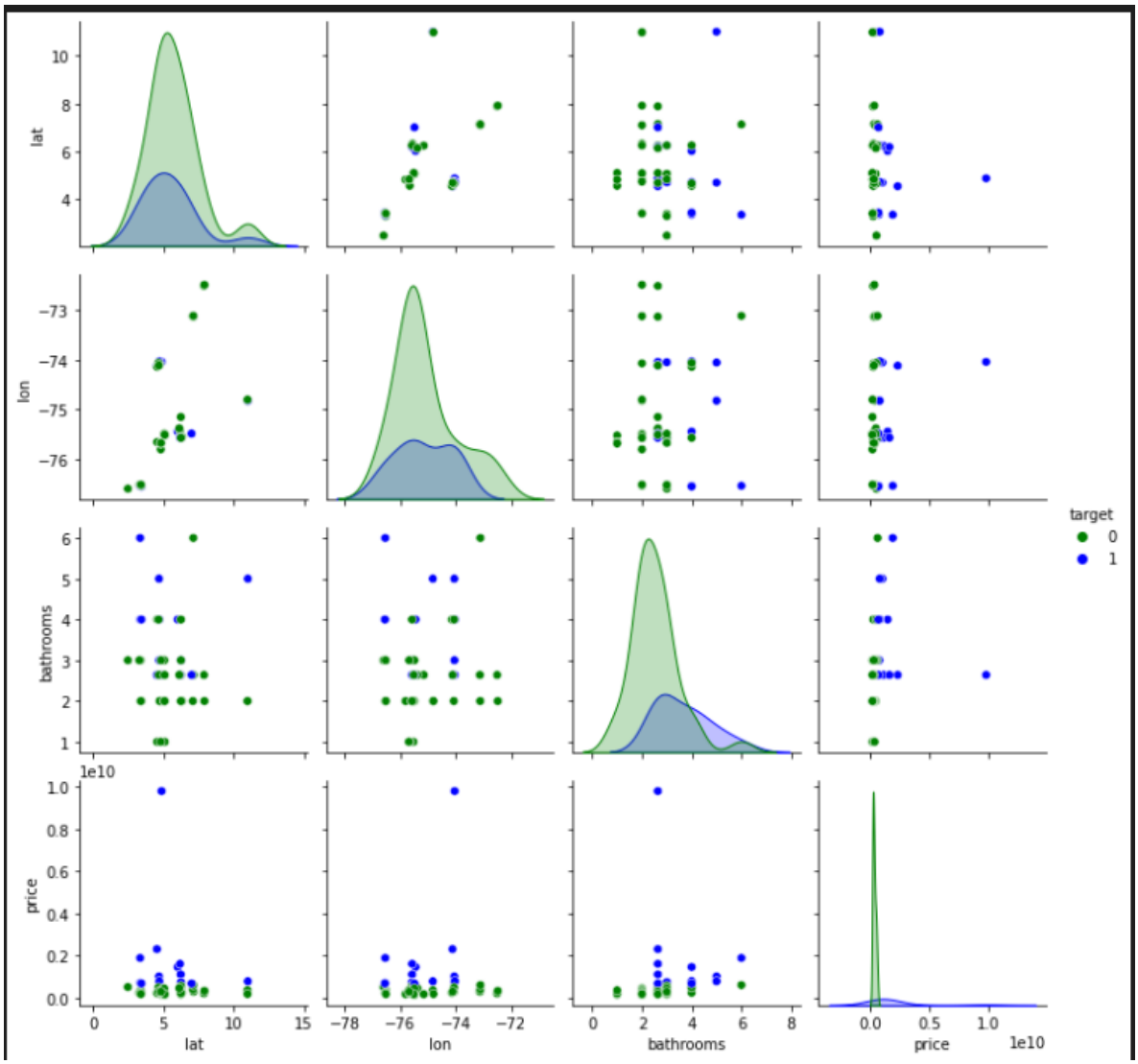


- Gracias al grafico me di cuenta facilmente que habia muchas columnas con gran porcentaje de valores nulos, Por ejemplo: l4,l5,l6,rooms,bed_rooms,surface_total,etc. Todas las columnas nombradas anteriormente tenian un porcentaje de nulos superior al 80%. Cualquiera pensaria que lo mas facil seria eliminarlas pero en esta tabla no fue esa una opcion posible ya que muchas de esas columnas estaban altamente correlacionadas con la columna de precios del modelo, por lo cual nos iban a ser de gran utilidad mas adelante. Por esta razon considere que una buena forma de realizar este proceso seria la de rellenar esos valores faltantes, ahora la paradoja es con que datos rellenarlos. Pense que la mejor forma de rellenar las columnas que ibamos a usar para predecir el modelo era la de rellenar con el promedio de dichas columnas.

```
test_data.surface_total.fillna(test_data.surface_total.mean())
train_data.surface_total.fillna(train_data.surface_total.mean())
```

- Este proceso fue aplicado a practicamente todas las columnas las cuales nos servirian para predecir el modelo por tener una alta correlacion con la columna de precios, las cuales son:


Unnamed: 0
lat
lon
rooms
bedrooms
bathrooms
surface_total
surface_covered
price
0.089
0.0069 0.35
0.0074 -0.086 -0.088
0.0089 -0.084 -0.052 1
0.0066 -0.077 -0.029 0.65 0.58
0.01 -0.0067 -0.00064 0.073 0.06 0.09
-0.012 -0.0089 -0.015 -0.0052 -0.004 0.0017 0.013
-0.0075 -0.0048 0.033 0.18 0.12 0.44 0.41 0.21
1.0
0.8
0.6
0.4
0.2
0.0



Correlacion:

rooms 0.178329

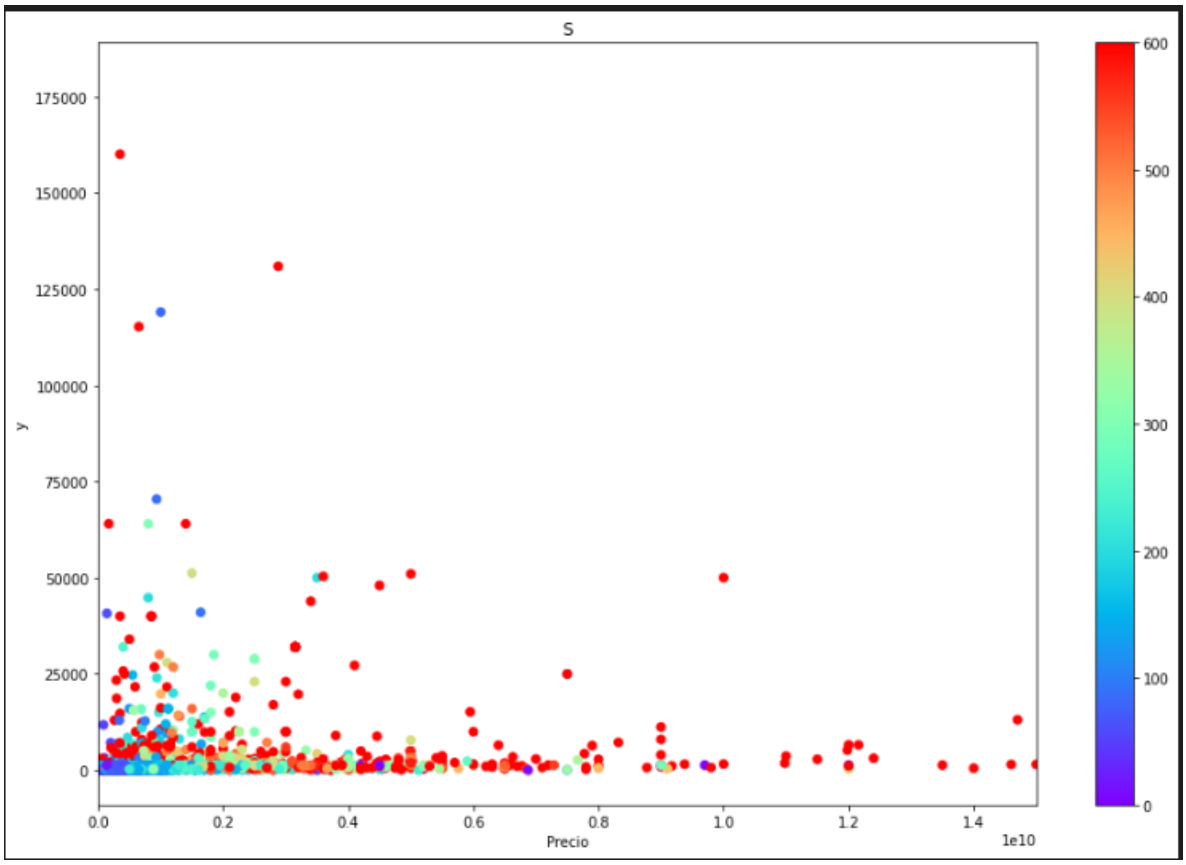
bedrooms 0.121947

bathrooms 0.443496

surface_total 0.414640

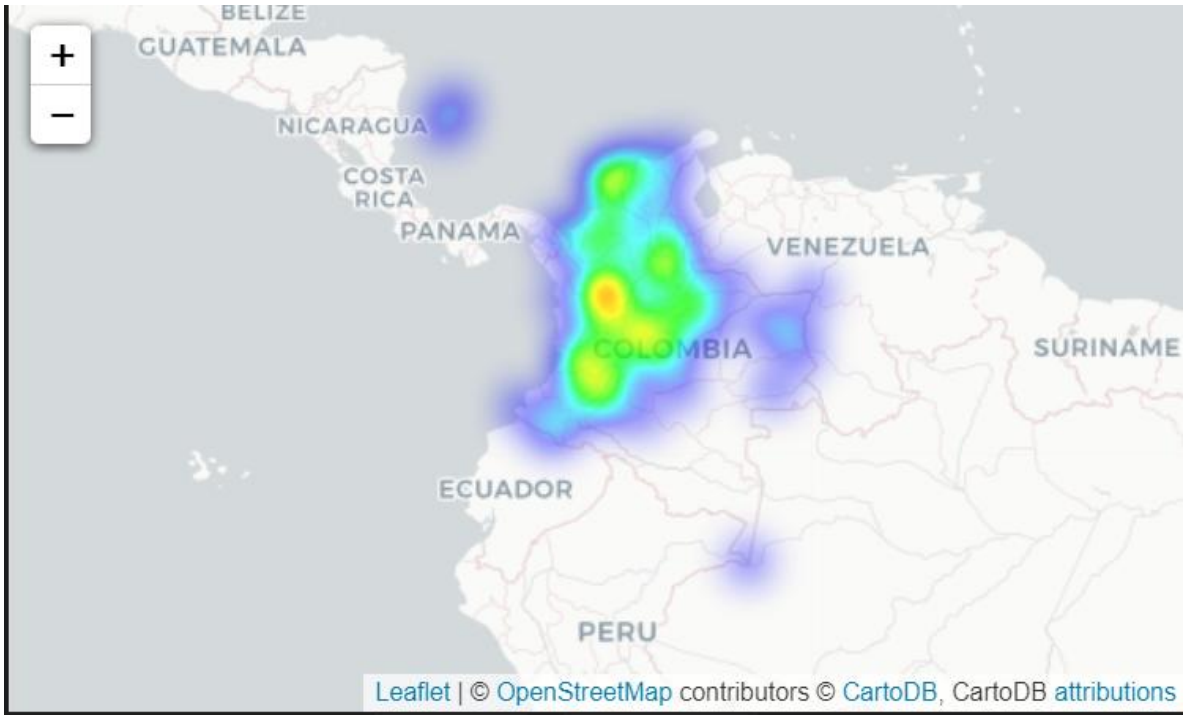
surface_covered 0.206906

- Aca podemos ver la alta correlacion que tienen estas columnas con la columna de precios, por ejemplo la columna de surface_total es la segunda columna mas correlacionada del modelo, por lo que por mas de que dicha columna tenga mas de un 90% de valores nulos no es posible simplemente borrarla por su alta correlacion.



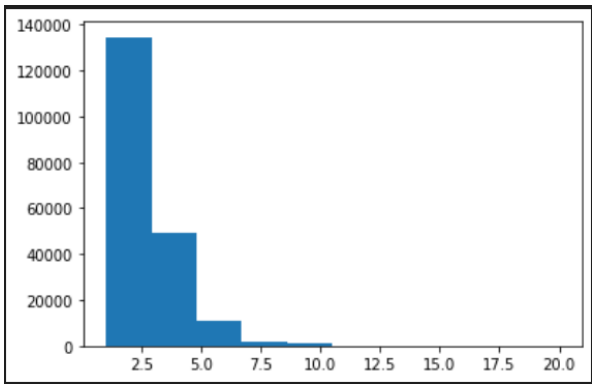
En el grafico anterior se puede observar la correlacion de la columna de precio con la columna de surface_total.

- Luego nos damos cuenta en base a las columnas de latitudes y longitudes que hay algunos valores de propiedades los cuales son valores atipicos ya que no se consideran dentro del pais de colombia. En base a nuestro modelo nos podemos dar cuenta de dos propiedades las cuales se encuentran en Chile y en Estados Unidos de America.
- Si bien estas propiedades no se consideran un gran problema para la prediccion de nuestro modelo ya que 2 propiedades en un modelo de mas de 100.000 registros presenta una mejora practicamente nula, de iugal forma decidi borrarlas ya que hacen que nuestro modelo no sea estetico al visualizarse en un mapa.
- Para solucionar este modelo cree dos funciuones que rellenan los valores nulos de las columnas de latitud y longitud y aparte que borren aquellos valores atipicos que se encuentran fuera de un radio de 150km.

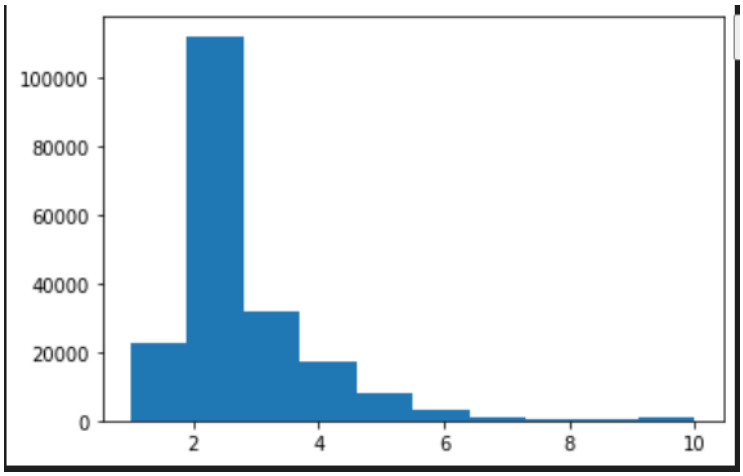


- Aca se puede ver como las unicas propiedades que quedaron en nuestro modelo son las que se encuentran en el pais de Colombia.

- Luego un paso en nuestro modelo fue la deteccion de valores atipicos, especificamente en la columna de bathrooms.



- Este grafico representa la cantidad de baños en cada propiedad, por lo que se puede ver hay algunas propiedades las cuales tiene mas de 10 baños, ya que en este modelo no solo se consideran inmuebles de vivienda sino tambien inmuebles comerciales o edificios en los cuales es normal tener un numero elevado de baños.
- Para la normalizacion de este aspecto decidi truncar los valoresy aquellas propieades que posean mas de 10 baños se quedan solo en 10 baños.
- Posteriormente se puede ver mismo grafico anterior pero luego de realizar una funcion para la normalizacion de la columna bathrooms.



- Luego agregamos una columna llamada target relacionada con la columna de precios. Esta nueva columna tiene informacion binaria de acuerdo a si la propiedad es cara o barata. En caso de que la propiedad supere la media de precio, se condiera cara, en caso contario esta se considera barata.

- Luego realizamos un borrado de columnas de acuerdo a lo que nos sirve para predecir el modelo, ya que hay muchas columnas que no nos interesan para esta parte de la prediccion:

```
COLUMNAS QUE SE BORRAN.
Unnamed: 0   :no aporta informacion para predecir, es un indice.
   id             :no aporta informacion para predecir, es un indice.
   ad_type        :no aporta informacion, todo el mismo valor.
   start_date     :no aporta informacion para predecir, es una fecha.
   end_date       :no aporta informacion para predecir, es una fecha.
   created_on     :no aporta informacion para predecir, es una fecha.
   l1             :Misma informacion en lat y long, ademas es monovalor).
   l2             :Misma informacion en lat y long.
   l3             :Misma informacion en lat y long.
   l4             :Misma informacion en lat y long.
   l5             :Misma informacion en lat y long.
   l6             :Misma informacion en lat y long.
   rooms          :Muchos nulos.
   bedrooms       :Muchos nulos.
   surface_total  :Muchos nulos.
   surface_covered:Muchos nulos.
   price          :De acá se genera la variable target.
   currency       :no aporta informacion, todo el mismo valor.
   price_period   :muchos nulos, y monovalor que no tiene sentido al ser una operacion de
venta.
   title          :por ahora no extraemos info de la misma.
   description    :no se considera directamente, se representa con la info extraida en col
desc_tot_words.
   operation_type :no aporta informacion, todo el mismo valor.
   geometry       :Misma informacion en lat y long.
   caro           :VAR target s/consigna 1 = caro / 0 = barato.
```

- Luego de esto realizamos un escalado de las variables para considerarlas en el modelo
- Realizamos una funcion para cambiar la codificacion de las variables
- Luego empezamos a realizar las predicciones correspondientes al modelo,. Me deicid por utilizar un arbol de decision ya que tenemos muchas variables con alta correlacion con la columna de target por lo que este modelo nos es de gran utilidad para que automaticamente elija cuales columnas ir usando para la prediccion.
- Luego de esto simplemente entrenamos el modelo con las columnas ya elejidas anteriormente e ir guardando las predicciones en una variable aparte.
- Luego vemos que tan preciso es nuestra prediccion la cual nos da los siguientes valores:

  Accuracy: 0.9000962074029065

  Recall: 0.7798369162342476

  F1 Score: 0.7886901574381493

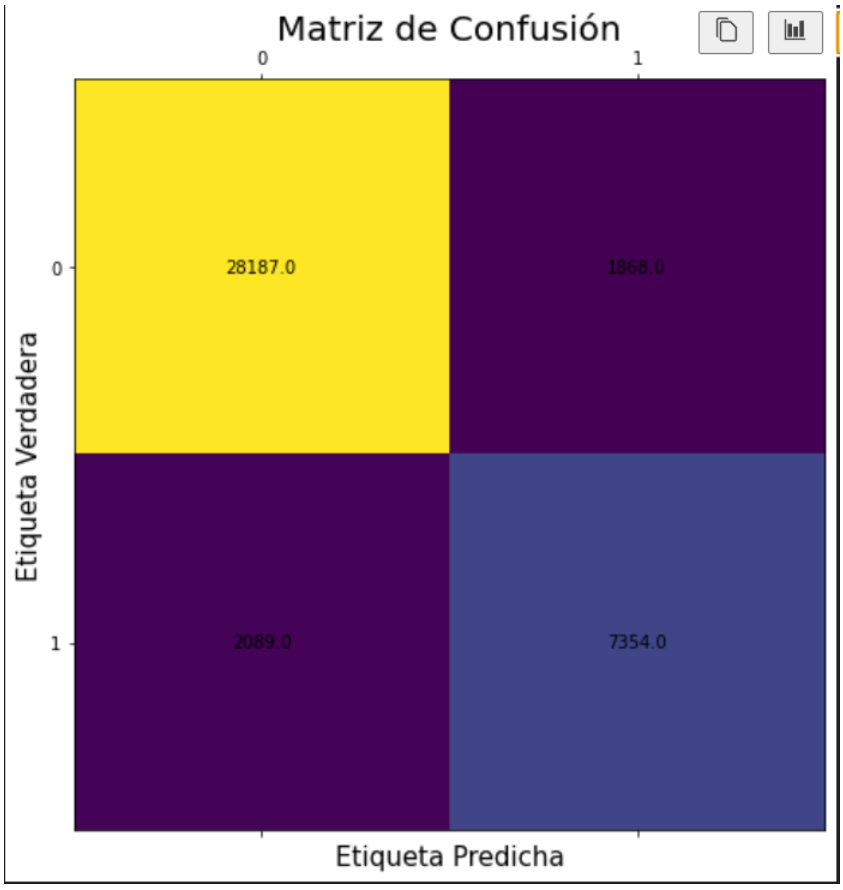
- Esto explica que el acuracy del modelo es de un 90%, el recall un 77% y el F1 Score un 78%.

- Luego vemos el porcentaje de error que tiene nuestro modelo el cual es el siguiente:

  Error en datos de test: 0.10018228771077016

- Esto explica que nuestro modelo posee un porcentaje de error de un 10%.

- Para finalizar realizamos un grafico de la Matriz de confusion para las mismas intenciones de los pasos anteriores, ver quue tan buena es nuestra prediccion.



Pasos Extra :

- Para finalizar este trabajo realice en un archivo aparte llamado 'prediccion_fechas.ipynb' una prediccion para los proximos meses de los precios de las propiedades del modelo. Para esto primero tuve que normalizar la columna de fecha de el archivo. Para este trabajo de normalizacion tuve que primero borrar aquellas fechas atipicas ya que habian algunas fechas en las cuales tenia un año inexistente los cuales tenia de años 9999.
- Luego entrenamos nuestro modelo para realizar las predicciones de los meses posteriores, en eSte caso utilizamos un modelo ARIMA para realizar la prediccion. Esto es simplemente para darnos una idea no es un trabajo muy preciso ya que lo que hace este modelo simplemente es ver la variacion de los precios por cada semana y en base a esto realiza una prediccion para los proximos meses pero no se consideran ningunas variables apartes como nuestro otro modelo el cual si es preciso por contener muchas mas variables aparte que solamente el precio. En un futuro se podria realizar un trabajo mucho mas fino para predecir las fechas de las proximas semanas y meses considerando mas varibales asi como la superficie, etc.


Modelo ARIMA -estacional- para predecir los precios de las propiedades en los proximos meses en colombia
1e9
price
prediccion
1.4
1.2
1.0
0.8
0.6
0.4
2020-01
2020-04
2020-07
2020-10
2021-01
2021-04
2021-07
2021-10
2022-01
2022-04