PREDICCIÓN ECONÓMICA DE VIVIENDAS

CARLOS ALBERTO BOLAÑOS

ID 1004134214

YEIMY TATIANA LÓPEZ GUERRERO

ID 1214746886

ALEJANDRO GOMEZ BORJA

1035861899

INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

FACULTAD DE INGENIERÍA

UNIVERSIDAD DE ANTIOQUIA

2022-1

Dadas las características de una vivienda (zona de clasificación, dimensiones, forma, acceso, etc), se predecirá el precio de venta de esta en el mercado. Se implementará el dataset de kaggle (https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data), que tiene 1459 muestras (casas) y columnas como: SalePrice, LotArea, HouseStyle, Exterior1st, Calefacción, entre otras.

Como métrica de Machine Learning se usará la raíz del error cuadrático medio (RMSE) entre el logaritmo del valor predicho y el logaritmo del precio de venta observado. (Tomar registros significa que los errores al predecir casas caras y casas baratas afectarán el resultado por igual). Si el error de los precios predichos por el algoritmo creado es superior al 20% en comparación con los precios reales de las casas, el modelo no será puesto en producción, ya que tendría poca confiabilidad para predecir y por ende, las ventas de las casas no tendrían un aumento notable.

1. Planteamiento del problema

Para las compañías que desempeñan sus labores comerciales con propiedad raíz, tal como la venta y alquiler de casas, es importante conocer el precio de venta de una propiedad tanto para la empresa y el cliente, ya que se tendrá de entrada se conocerá el presupuesto con el cual se debe contar a la hora de cerrar un trato. Es por todo esto, que se desea desarrollar un modelo que permita predecir el precio final de una propiedad teniendo en cuenta todas las variables que determinan ese precio final.

Dataset

El dataset proviene de una competencia de Kaggle, donde se proporcionan datos históricos de más de 1400 casas en los Estados Unidos, con múltiples variables tales como: Área, localización, vecindario, dormitorios, utilidades, año de construcción, comodidades, y muchas más). El dataset viene con archivos .csv divididos en train y test, los cuales sirven para entrenar el modelo de predicción y probarlo respectivamente.

Métrica

Como métrica de Machine Learning se usará la raíz del error cuadrático medio (RECM). La RECM es la raíz cuadrada del promedio de errores cuadrados. El efecto de cada error en la RECM es proporcional al tamaño del error cuadrado; por lo tanto, los errores mayores tienen un efecto desproporcionadamente grande en la RECM. Por lo tanto, la RECM es sensible a los valores atípicos.

Se calcula mediante la siguiente expresión:

$$RECM = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}}$$

La RECM de los valores predichos $\hat{y}$ para t veces la regresión de la variable dependiente yt con variables observadas T veces, se calcula para T diferentes predicciones como la raíz cuadrada de la media de los cuadrados de las desviaciones.

Variable objetivo

La variable objetivo que se desea predecir, es la última columna de los datos train.csv, la cual corresponde a 'SalePrice'. Se estudiará cuáles son las variables principales que se relacionan con el resultado objetivo.

1.1 Librerías

A continuación se procede a mostrar las librerías que se usaron y la carga de archivos correspondientes.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
```

Fig 1. Librerías.

1.2 Archivos a utilizar

Se procede a realizar la carga de los datos de entrenamiento train.csv y prueba test.csv para el modelo de predicción, en este caso se muestra en la Figura 2 las primeras 5 filas de los datos de entrenamiento:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType	SaleCondition	SalePrice
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2008	WD	Normal	208500
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	5	2007	WD	Normal	181500
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	9	2008	WD	Normal	223500
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2006	WD	Abnorml	140000
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	12	2008	WD	Normal	250000

5 rows × 81 columns

Fig 2. Datos de entrenamiento.

2. Análisis de datos

En este apartado se hace un análisis profundo sobre el comportamiento de la variable objetivo, distribuciones, relaciones con las demás variables independientes. También, se hace la reducción de variables que no influyan considerablemente en el modelo, el filtrado y limpieza general de los datos a trabajar.

2.1 Distribución de datos

Ya que la variable que se intenta predecir es "SalePrice", se informa sobre ella usando el método describe (), el cual, entrega datos relevantes, como lo son: Cantidad de datos, promedio, desviación estándar, etc.

Ya que la variable que interesa analizar es el precio de venta "SalePrice", se verifica que no tenga valores vacíos, en caso tal de que los haya, se remueven.

```
b=datos['SalePrice'].describe()
print(b)
```

```
count      1460.000000
mean     180921.195890
std       79442.502883
min       34900.000000
25%      129975.000000
50%      163000.000000
75%      214000.000000
max      755000.000000
Name: SalePrice, dtype: float64
```

Fig 3. Descripción SalePrice.

Para ver la forma (distribución) de los datos del precio de venta de las casas, se procede a graficar un histograma:

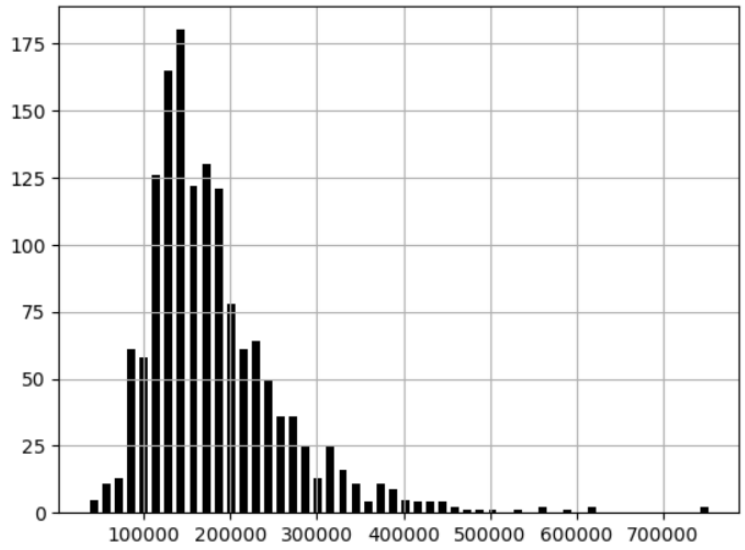

Fig 4. Distribución de datos SalePrice.

Ya que hay muchas variables en este ejercicio, se procede a ver las variables que tengan una mayor correlación con SalePrice, para así, tener a estas en cuenta a la hora de proceder al entrenamiento para predecir el precio. Para ver la correlación de las variables, se crea una matriz de correlación.

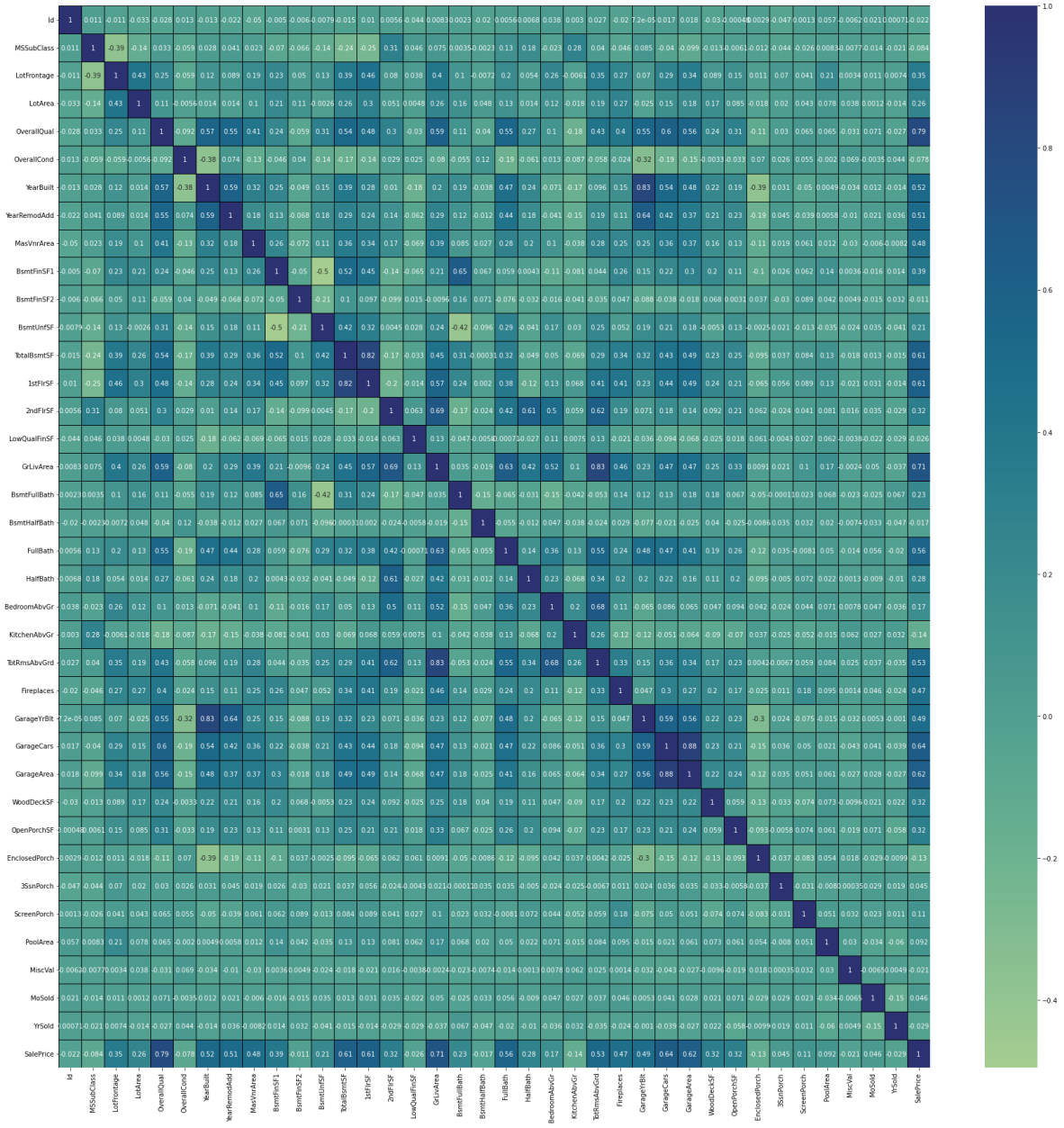

Fig 5. Matriz de correlación.

Con este diagrama de calor se puede apreciar las relaciones que tiene SalePrice con las demás categorías, apreciando celdas de diferente color debido a que estas representan niveles de relación entre variables. Entre más intenso sea el color hay una mayor relación.

Con la siguiente línea de código realizamos un nuevo filtro para encontrar cuáles variables tienen una mayor relación con SalePrice dependiente del valor que deseemos.

Con la siguiente línea de código realizamos un nuevo filtro para encontrar cuáles variables tienen una mayor relación con SalePrice dependiente del valor que deseemos.

```
x=c["SalePrice"][c["SalePrice"]>0.5]
print(x)
```

```
OverallQual     0.790982
YearBuilt       0.522897
YearRemodAdd    0.507101
TotalBsmtSF     0.613581
1stFlrSF        0.605852
GrLivArea       0.708624
FullBath        0.560664
TotRmsAbvGrd    0.533723
GarageCars      0.640409
GarageArea      0.623431
SalePrice       1.000000
Name: SalePrice, dtype: float64
```

Fig 6. DataFrame de correlación

Tomamos un valor de correlación mínimo de 0.5 y en la lista “x” se presentan algunas de las categorías relacionadas que cumple esta condición.

3. Gráfica de variables correlacionadas

Se grafica la variable SalePrice junto con las variables que más se correlacionan a ella, tal como se podrá observar, todas estas presentan una correlación positiva, es decir, si una de las variables aumenta, la otra también lo hace:

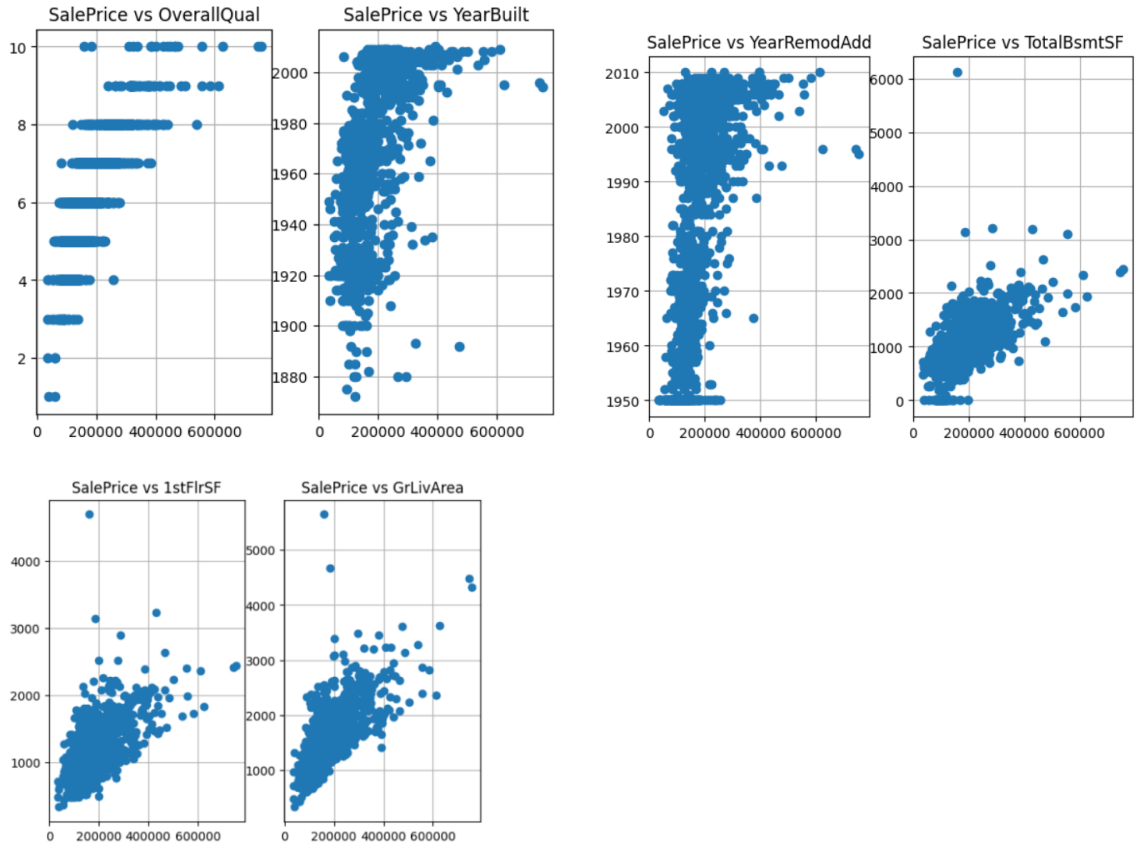

Fig 7. Gráficas de correlación.

Se puede observar la variable “SalePrice” en el eje de las x y la otra variable relacionada en el eje y. De izquierda a derecha, disminuye el nivel de correlación de Pearson, siendo la variable independiente “OverallQual” la que más incide en el precio final de la vivienda.

3.1 Datos vacíos

A continuación, se informa acerca de la cantidad de datos vacíos por cada variable independiente y el porcentaje equivalente respecto a la cantidad total de datos respectivamente. Esto para los datos de entrenamiento y prueba.

Se llenan los datos vacíos de mayor cantidad con el promedio asociado a la respectiva variable.

	Cant. Nulos	Porcentaje
OverallQual	0	0.0
YearBuilt	0	0.0
YearRemodAdd	0	0.0
TotalBsmtSF	0	0.0
1stFlrSF	0	0.0
GrLivArea	0	0.0
FullBath	0	0.0
TotRmsAbvGrd	0	0.0
GarageCars	0	0.0
GarageArea	0	0.0
SalePrice	0	0.0

Fig 8. Datos nulos de entrenamiento.

	Cant. Nulos	Porcentaje
TotalBsmtSF	1	0.000685
GarageCars	1	0.000685
GarageArea	1	0.000685
OverallQual	0	0.000000
YearBuilt	0	0.000000
YearRemodAdd	0	0.000000
1stFlrSF	0	0.000000
GrLivArea	0	0.000000
FullBath	0	0.000000
TotRmsAbvGrd	0	0.000000

Fig 9. Datos nulos de prueba.

Los datos observados en las tablas 8 y 9 representan una lista de datos nulos que se presentan en nuestro DataFrame, pero como se puede ver algunos tienen un porcentaje nulo y otros son demasiado pequeño, es decir, no hay necesidad de eliminarlos, ya que no presentarán problema alguno para una buena predicción.

3.2 Normalización

Tal como se verá a continuación en la Figura 10.1 y 10.2 los datos de la columna SalePrice del DataFrame, no se pueden representar mediante una distribución de probabilidad normal:

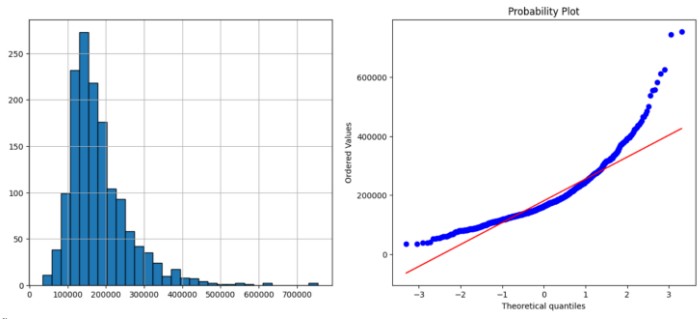

Fig 10.1 y Fig 10.2. Función de probabilidad.

En las figuras 10.1 y 10.2, los datos de la columna SalePrice del DataFrame (gráfico azul), no siguen la distribución de probabilidad normal (gráfico rojo), por tanto, para tener un mejor entrenamiento de la variable, se debe encontrar una distribución que represente bien el comportamiento de los datos. Se procede a realizar un código para transformar distribuciones con sesgo positivo (como es nuestro caso), la transformación logarítmica es la más usada, puesto que en la escala logarítmica, la distancia es exactamente la misma entre 1 y 10 que entre 10 y 100 o 100 y 1000, etc. Lo que resulta en que la parte izquierda se expandirá, mientras que la parte derecha se comprimirá, lo que favorecerá a la curva resultante para que se ajuste mejor a una normal. En la Figura 11.1 y 11.2 se observa la normalización llevada a cabo.

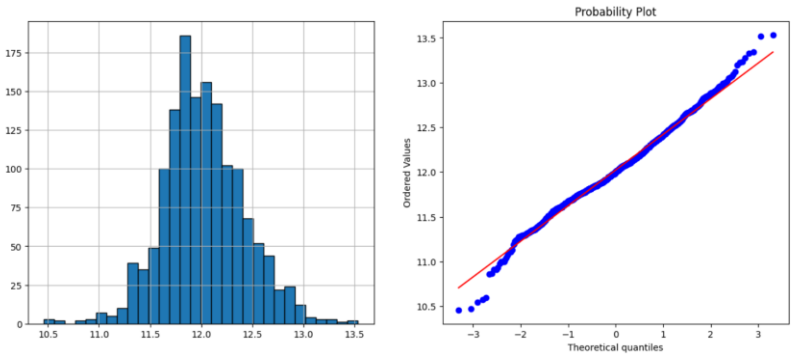

Fig 11.1 y Fig 11.2. Función probabilidad normalizada.

3.3 Datos Dummis

Las variables Dummies o indicadoras sirven para identificar las categorías a las cuales pertenecen las observaciones, en este caso si pertenecen al precio de venta de vivienda mediante los valores de 0 o de 1. Por tanto, transformaremos el DataFrame que se tiene en el momento (sin datos faltantes) a variables Dummies.

Se verifica si quedó algún dato faltante en los datos de prueba:

	Porcentaje Faltante
TotalBsmtSF	0.06854
GarageCars	0.06854
GarageArea	0.06854

Fig 12. Datos de prueba.

Esto concuerda según la Figura 9, donde se tenía 1 dato vacío por cada variable faltante respectivamente. Estos datos vacíos, al ser solo 1 por cada variable, se procede a llenar con un cero.

4. Modelos:

Selección de datos

En esta parte del proyecto se realiza la creación de un DataFrame que contenga datos aleatorios de las dos listas que se usaron: train y test.

```
[220000 220000  75500 ... 129900 183900 372500]
      OverallQual  YearBuilt  YearRemodAdd  TotalBsmtSF  1stFlrSF  GrLivArea  \
133             8       2001          2002         1267      1296       1296
162             7       2005          2005         1541      1541       1541
614             4       1972          1972          630       630        630
939             7       1940          1950         1032      1207       2403
586             6       1918          2000          816       838        838
...           ...        ...           ...          ...       ...        ...
631             8       2006          2006         1554      1554       1554
163             4       1956          1956          882       882        882
8               7       1931          1950          952      1022       1774
1265            7       1999          1999          691       713       1452
678             8       2008          2008         2046      2046       2046

      FullBath  TotRmsAbvGrd  GarageCars  GarageArea
133          2             6           2         471
162          2             7           2         532
614          1             3           0           0
939          2            10           1         349
586          1             5           1         275
...        ...           ...         ...         ...
631          2             6           2         627
163          1             4           0           0
8            2             8           2         468
1265         2             6           2         506
678          2             7           3         834

[1168 rows x 10 columns]
```

Fig 13. Datos seleccionados para la prueba.

4.1 Modelo lineal

Con la ayuda de la función “predict” se realiza la predicción de precios de ventas de casas teniendo diferentes parámetros tomados de los documentos.

```
[206213.04347826 121855.38349515 153726.27619048 121855.38349515
 212870.         442657.04347826 121855.38349515 121855.38349515
 121855.38349515 212870.         184456.84172662 121855.38349515
  93285.24691358 121855.38349515 121855.38349515 121855.38349515
  93285.24691358  93285.24691358 121855.38349515 333837.6875
 153726.27619048 121855.38349515 180749.97142857  93285.24691358
 121855.38349515 121855.38349515 155154.8        311609.
 273689.86046512 153726.27619048 155154.8        130946.92307692
 180749.97142857 153726.27619048 121855.38349515 184750.
 153726.27619048 240557.         155154.8        121855.38349515
 212870.         442657.04347826 273689.86046512 180749.97142857
 121855.38349515 121855.38349515 273689.86046512 121855.38349515
 130946.92307692 184456.84172662 180749.97142857 121855.38349515
 121855.38349515 153726.27619048 180749.97142857 244891.47619048
 180749.97142857  93285.24691358 333837.6875     180749.97142857
 184456.84172662 180749.97142857 121855.38349515 131015.55555556
 244891.47619048 131015.55555556 442657.04347826 121855.38349515
 212870.         121855.38349515 240557.         212870.
 153227.4         93285.24691358 180749.97142857 311609.
 184456.84172662 121855.38349515 212870.         290333.33333333
 121855.38349515 180749.97142857 184456.84172662 121855.38349515
 121855.38349515 153726.27619048  93285.24691358 155154.8
 180749.97142857 257607.27272727 184456.84172662 130946.92307692
 153726.27619048 121855.38349515 121855.38349515 184456.84172662
 180749.97142857 131015.55555556 121855.38349515 305582.16666667
 153726.27619048 184456.84172662 130946.92307692 121855.38349515
 121855.38349515 184456.84172662 155154.8        131015.55555556
 121855.38349515 184456.84172662 153726.27619048 155154.8
 228697.64705882 184456.84172662  93285.24691358 153726.27619048
  93285.24691358 155154.8        184456.84172662  93285.24691358
 273689.86046512 333837.6875     121855.38349515 180749.97142857
 121855.38349515 240557.         121855.38349515  93285.24691358
 153726.27619048 180749.97142857 184456.84172662 240557.
 121855.38349515  93285.24691358 442657.04347826 275434.5
 257607.27272727 184456.84172662 153726.27619048 121855.38349515
```

Fig 14. Tabla con los valores predecidos.

4.2 DataFrame: datos de predicción

Finalmente, se convierte el resultado obtenido a un dataframe para mejor visualización:

	Id	SalePrice
0	1461	206213.043478
1	1462	121855.383495
2	1463	153726.276190
3	1464	121855.383495
4	1465	212870.000000
5	1466	442657.043478
6	1467	121855.383495
7	1468	121855.383495
8	1469	121855.383495
9	1470	212870.000000
10	1471	184456.841727
11	1472	121855.383495
12	1473	93285.246914
13	1474	121855.383495
14	1475	121855.383495
15	1476	121855.383495
16	1477	93285.246914
17	1478	93285.246914
18	1479	121855.383495
19	1480	333837.687500

Fig 15. DataFrame valor predecidos.

5. Retos y consideraciones de despliegue

Para poder evaluar el desempeño del modelo, se mide el error de los precios predichos por el algoritmo creado, si este error es superior al 20% en comparación con los precios reales de las casas, el modelo no será puesto en producción, ya que tendría poca confiabilidad para predecir y por ende, las ventas de las casas no tendrían un aumento notable. Si el modelo permite tener un ahorro o un mayor control del precio de las viviendas por parte de las compañías de bienes raíces, el modelo podría estar listo para desplegarse. Es necesario estar alimentando el modelo, ya sea con datos almacenados en una nube, que permita estar en constante ajuste al modelo, o en constaste entrenamiento del modelo, dado el caso que el error supere el porcentaje anteriormente mencionado.

6. Conclusiones

- Se hace necesario realizar un análisis detallado de los datos, un buen filtrado y correcta eliminación de posibles datos erróneos que puedan afectar el resultado final.
- Es importante conocer la posible distribución del resultado esperado, clasificar correctamente cada variable independiente y tratar siempre encontrar las variables más influyentes en la predicción.
- Los modelos de decisión son ofrecen ventajas respecto a otros, en el sentido de que permiten medir otro tipo de relaciones entre variables, y el parámetro no solamente puede ser numérico, también puede ser categórico.