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

ENTREGA 2

Inicialmente se procede a cargar los datos en el notebook para comenzar la predicción de alguna de las variables que contiene el dataset “precio de la vivienda”, para escoger cuál variable vamos a analizar se piensa en algún problema general que se presente en nuestro diario vivir; en este caso, notamos que es de gran relevancia saber el costo de la vivienda, ya que esto está entre las cosas que más le interesa al comprador y vendedor al momento de ofrecer un inmueble.

A continuación, se presenta la tabla de datos obtenida mediante la importación de librerías que con sus funciones se logra la correcta lectura del archivo csv llamado Train.

```
import pandas as pd
import matplotlib.pyplot as plt
#import seaborn as sns

datos=pd.read_csv('train.csv',header=0)
print(datos)
datos.tail()
```

```
         Id  MSSubClass MSZoning  LotFrontage  LotArea Street Alley LotShape  \
0         1          60       RL         65.0     8450   Pave   NaN      Reg
1         2          20       RL         80.0     9600   Pave   NaN      Reg
2         3          60       RL         68.0    11250   Pave   NaN      IR1
3         4          70       RL         60.0     9550   Pave   NaN      IR1
4         5          60       RL         84.0    14260   Pave   NaN      IR1
...     ...         ...      ...          ...      ...    ...   ...      ...
1455   1456          60       RL         62.0     7917   Pave   NaN      Reg
1456   1457          20       RL         85.0    13175   Pave   NaN      Reg
1457   1458          70       RL         66.0     9042   Pave   NaN      Reg
1458   1459          20       RL         68.0     9717   Pave   NaN      Reg
1459   1460          20       RL         75.0     9937   Pave   NaN      Reg

     LandContour Utilities  ... PoolArea PoolQC  Fence MiscFeature MiscVal  \
0            Lvl    AllPub  ...        0    NaN    NaN         NaN       0
1            Lvl    AllPub  ...        0    NaN    NaN         NaN       0
2            Lvl    AllPub  ...        0    NaN    NaN         NaN       0
3            Lvl    AllPub  ...        0    NaN    NaN         NaN       0
4            Lvl    AllPub  ...        0    NaN    NaN         NaN       0
...          ...       ...  ...      ...    ...    ...         ...     ...
1455         Lvl    AllPub  ...        0    NaN    NaN         NaN       0
1456         Lvl    AllPub  ...        0    NaN  MnPrv         NaN       0
1457         Lvl    AllPub  ...        0    NaN  GdPrv        Shed    2500
1458         Lvl    AllPub  ...        0    NaN    NaN         NaN       0
1459         Lvl    AllPub  ...        0    NaN    NaN         NaN       0

     MoSold YrSold  SaleType  SaleCondition  SalePrice
0         2   2008        WD         Normal     208500
1         5   2007        WD         Normal     181500
2         9   2008        WD         Normal     223500
3         2   2006        WD        Abnorml     140000
4        12   2008        WD         Normal     250000
...     ...    ...       ...            ...        ...
1455      8   2007        WD         Normal     175000
1456      2   2010        WD         Normal     210000
1457      5   2010        WD         Normal     266500
1458      4   2010        WD         Normal     142125
1459      6   2008        WD         Normal     147500

[1460 rows x 81 columns]
```

Se puede apreciar en la imagen que el archivo train.csv contiene 1460 filas x 81 columnas, datos que cumplen las condiciones pedidas para el proyecto. Por cuestiones de espacio y cantidad de datos, no se puede apreciar cada uno de ellos, solamente las 5 primeras y 5 últimas filas.

La columna SalePrice contiene los precios de cada vivienda, entonces procedemos a extraer estos datos en otra lista que filtra esta información.

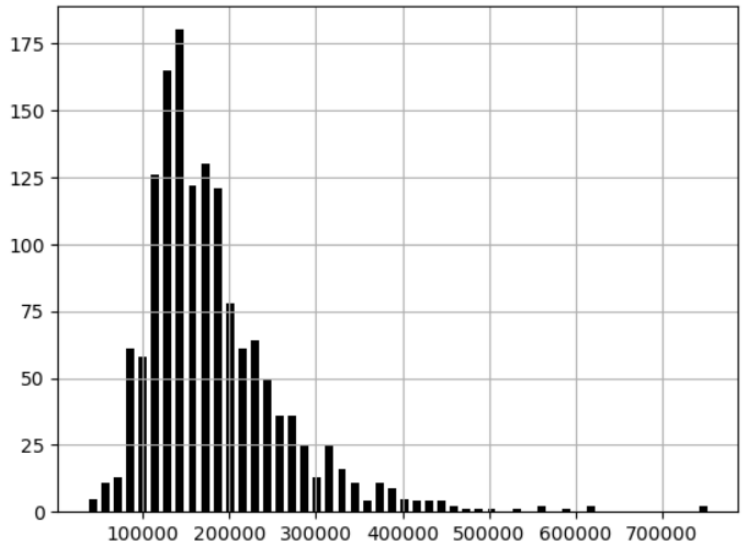
```
b=datos['SalePrice'].describe()
print(b)
```

```
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
```

Con la función .describe() podemos obtener las características de la columna, es decir, datos importantes como promedio, mínimo, máximo, percentiles, desviación estándar y número de datos.

Para una mejor visualización de los datos relacionados con el SalePrice procedemos a graficar un histograma, el cual es una gráfica de frecuencia vs precio de la casa.

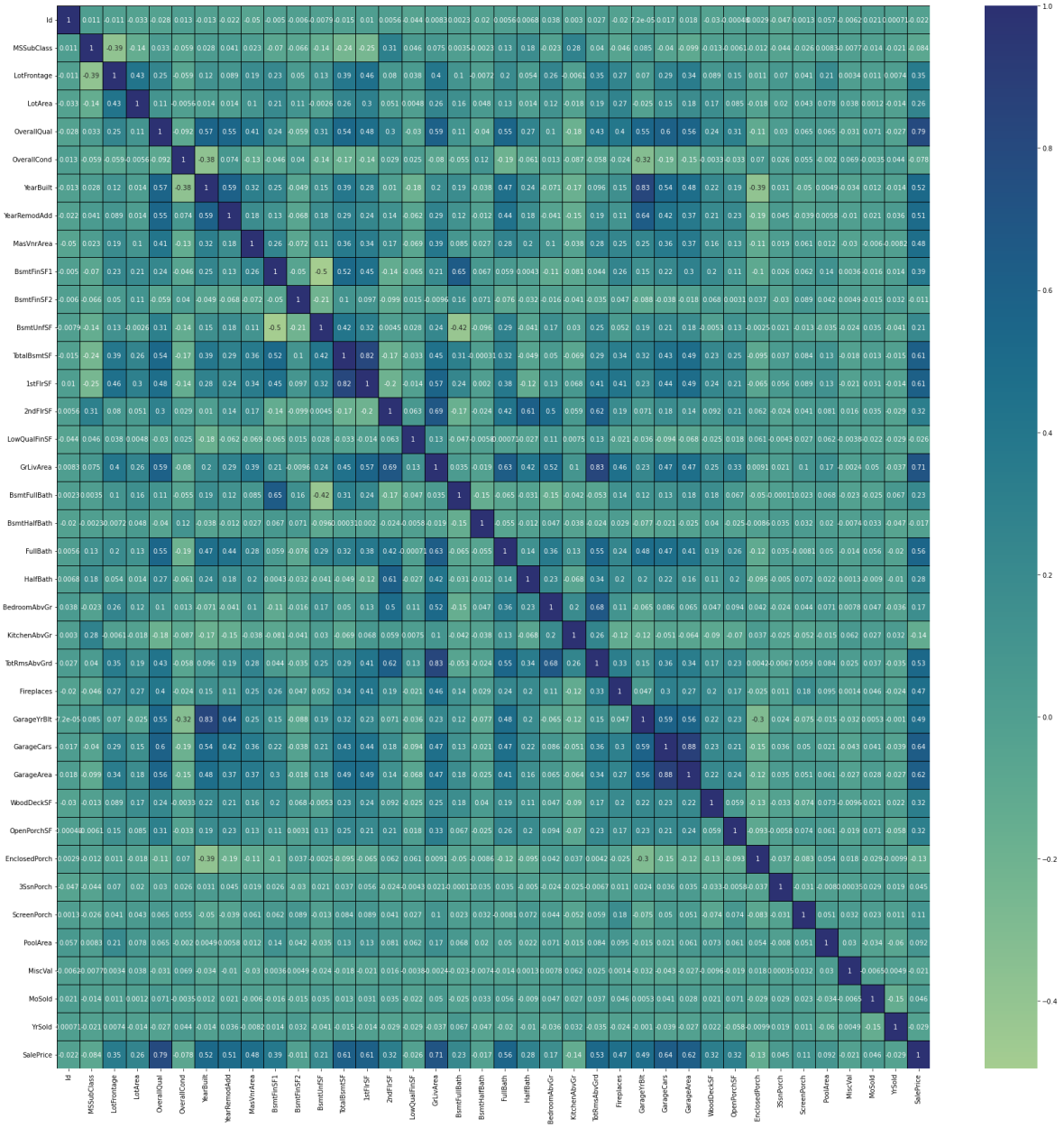
```
plt.hist(a,bins=50,rwidth=(0.65),color='k')
plt.grid()
```



Podemos ver en el histograma que la moda se encuentra concentrada entre precios de 100.000 y 200.000.

El precio de vivienda depende de varios factores como ubicación, tamaño, distribución y entre otras, entonces vemos la necesidad de realizar un filtro para relacionar SalePrice con otras categorías que contiene el archivo train. Procedemos a realizar un código para la matriz de correlación.

```
df=pd.DataFrame(datos)
c=df.corr(method='pearson')
plt.figure(figsize=(30, 30))
d=sns.heatmap(c, annot=True, cmap="crest")
plt.show()
```



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.

```
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
```

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.