Tercer parcial HPC - Juan Andrés González Molina

Contexto

Los siguientes procedimientos son llevados a cabo teniendo en cuenta el ambiente académico con el que se cuenta en esta materia, siendo procesos académicos exploratorios para profundizar en el proceso de regresión lineal e interiorizar los conceptos asociados a este proceso.

Para llevar a cabo esta práctica, fue provisto un dataset, en este caso un dataset con información sobre diamantes.

Modelos

Para esta práctica se realiza una regresión lineal sobre el dataset provisto.

El proceso de Regresión Lineal consiste en ajustar una línea recta que se ajuste a los datos minimizando lo que más se pueda el error, es decir, la distancia entre dicha línea recta y los datos.

Métricas

Existen distintas métricas para medir el rendimiento de un modelo, entre ellas:

R^2: Muestra qué tan bien los términos (puntos de datos) se ajustan a una curva o línea. El R2 ajustado también indica qué tan bien se ajustan los términos a una curva o línea, pero se ajusta para la cantidad de términos en un modelo. Si agrega más y más inútil variables a un modelo, el R cuadrado ajustado disminuirá. Si agrega más variables útiles, aumentará R cuadrado ajustado. R2 ajustado siempre será menor o igual a R^2.

```
[20]  1 # 1.0 -> Vectores de predicción de sklearn
      2 y_hat_train_sk = pipe.predict(X_train)
      3 y_hat_test_sk = pipe.predict(X_test)
```

```
[21]  1 # 2.1 -> Función para calcular la métrica R²
      2 from sklearn.metrics import r2_score
```

Métrica R^2 sobre

```
[22]  1 # 2.2 -> Métrica de rendimiento sobre los datos de entrenamiento.
      2 r2_score(y_train, y_hat_train_sk)
```

```
0.871558110664378
```

```
[23]  1 # 2.3 -> Métrica de rendimiento sobre los datos de entrenamiento.
      2 r2_score(y_test, y_hat_test_sk)
```

```
-0.9244472202090794
```

Conclusión sobre las métricas de rendimiento (R^2):

- Train: El modelo se ajusta de forma razonable a los datos de entrenamiento, teniendo un rendimiento que aunque no es notablemente preciso, maneja un nivel de precisión razonable.
- Test: El modelo no se ajusta de forma correcta a las predicciones, siendo incluso peor que calcular la media de los datos.

EDA

EDA: En estadística, el proceso del Análisis Exploratorio de Datos o EDA por sus siglas en inglés, es una serie de pasos para analizar sets de datos y extraer sus características principales, generalmente se utilizan visualizaciones o gráficas de funciones estadísticas que ayudan a ver las características intrínsecas de los datos que sirven de base para formular preguntas clave o hipótesis que más adelante traen valor

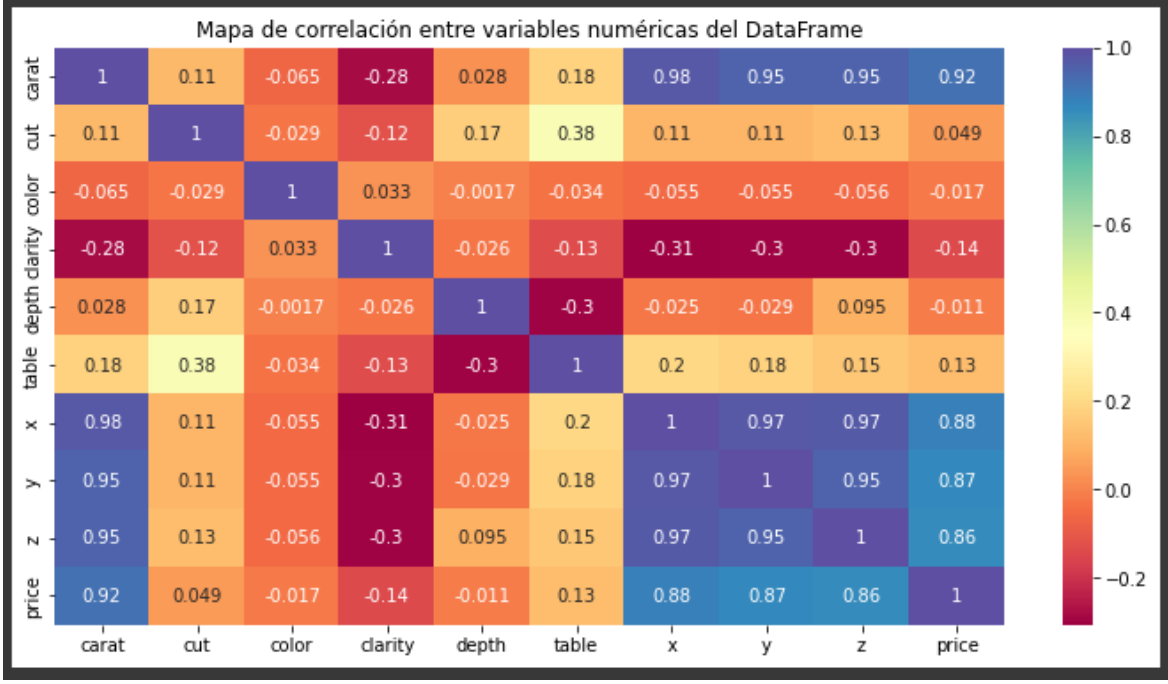
```
1 # EDA del dataset:
2
3 # 1.0 -> información completa del dataset
4 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 53940 entries, 0 to 53939
Data columns (total 10 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   carat    53940 non-null  float64
 1   cut      53940 non-null  int64
 2   color    53940 non-null  int64
 3   clarity  53940 non-null  int64
 4   depth    53940 non-null  float64
 5   table    53940 non-null  float64
 6   x        53940 non-null  float64
 7   y        53940 non-null  float64
 8   z        53940 non-null  float64
 9   price    53940 non-null  int64
dtypes: float64(6), int64(4)
memory usage: 4.1 MB
```

```
1 # 2.0 -> Se presenta un resumen estadístico de los datos.
2 df.describe()
```

	carat	cut	color	clarity	depth	table	x	y	z	price
count	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000
mean	0.797940	1.228940	3.174416	2.256136	61.749405	57.457184	5.731157	5.734526	3.538734	3932.799722
std	0.474011	1.265976	2.050156	1.766539	1.432621	2.234491	1.121761	1.142135	0.705699	3989.439738
min	0.200000	0.000000	0.000000	0.000000	43.000000	43.000000	0.000000	0.000000	0.000000	326.000000
25%	0.400000	0.000000	1.000000	1.000000	61.000000	56.000000	4.710000	4.720000	2.910000	950.000000
50%	0.700000	1.000000	4.000000	2.000000	61.800000	57.000000	5.700000	5.710000	3.530000	2401.000000
75%	1.040000	3.000000	5.000000	3.000000	62.500000	59.000000	6.540000	6.540000	4.040000	5324.250000
max	5.010000	4.000000	6.000000	7.000000	79.000000	95.000000	10.740000	58.900000	31.800000	18823.000000

```
1 # 3.0 -> A continuación se presenta la matriz de correlación: Relación entre
2 # las variables dependientes e independientes.
3
4 math_correlacion = df.corr()
5 plt.figure(figsize=(12,6))
6 sns.heatmap(math_correlacion, annot=True, cmap='Spectral')
7 plt.title('Mapa de correlación entre variables numéricas del DataFrame')
8 plt.show()
```



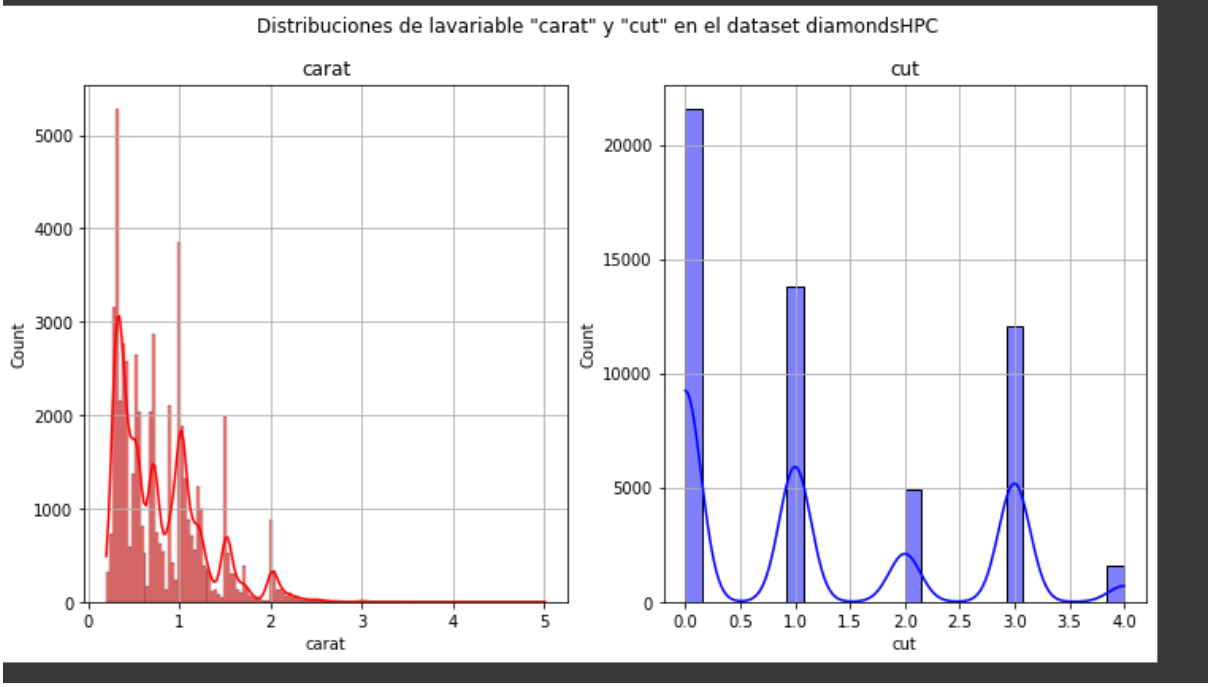
```python
from IPython.core.pylabtools import figsize
# 4.1 -> Se grafica la distribución de la variable independiente 'carat' y 'cut'
fig, axes = plt.subplots(1, 2, figsize=(12,6))

fig.suptitle('Distribuciones de lavariable "carat" y "cut" en el dataset diamondsHPC')

sns.histplot(df['carat'], ax=axes[0], kde=True, color='r')
axes[0].set_title('carat')
axes[0].grid()

sns.histplot(df['cut'], ax=axes[1], kde=True, color='b')
axes[1].set_title('cut')
axes[1].grid()

plt.show()
```



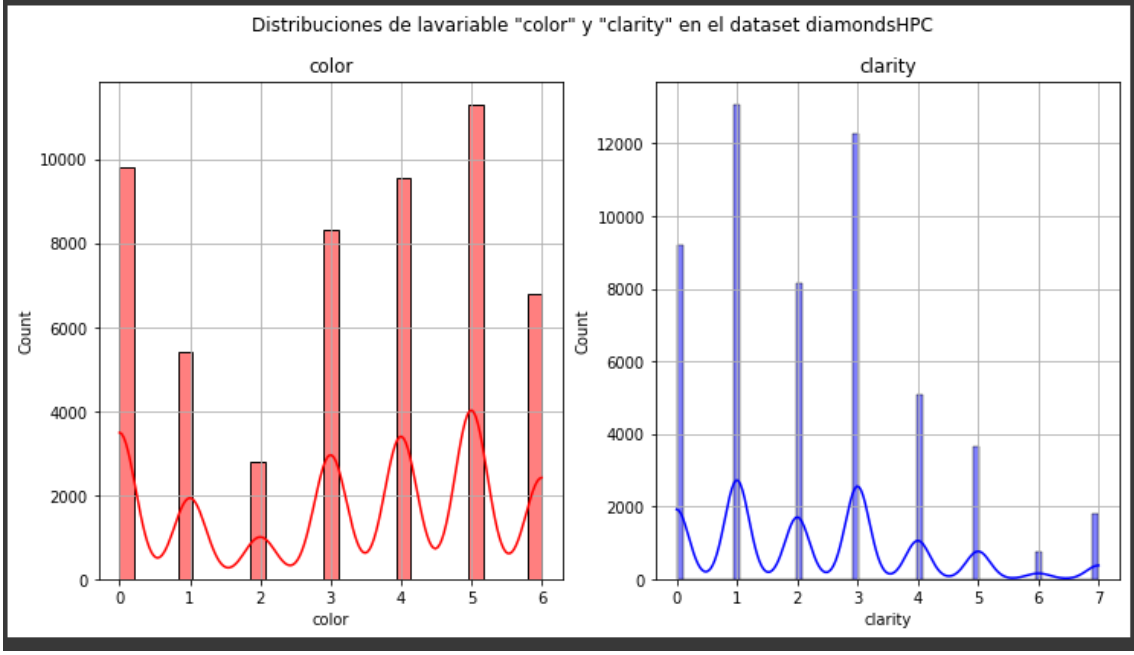
```python
from IPython.core.pylabtools import figsize
# 4.2 -> Se grafica la distribución de la variable independiente 'color' y 'clarity'
fig, axes = plt.subplots(1, 2, figsize=(12,6))

fig.suptitle('Distribuciones de lavariable "color" y "clarity" en el dataset diamondsHPC')

sns.histplot(df['color'], ax=axes[0], kde=True, color='r')
axes[0].set_title('color')
axes[0].grid()

sns.histplot(df['clarity'], ax=axes[1], kde=True, color='b')
axes[1].set_title('clarity')
axes[1].grid()

plt.show()
```



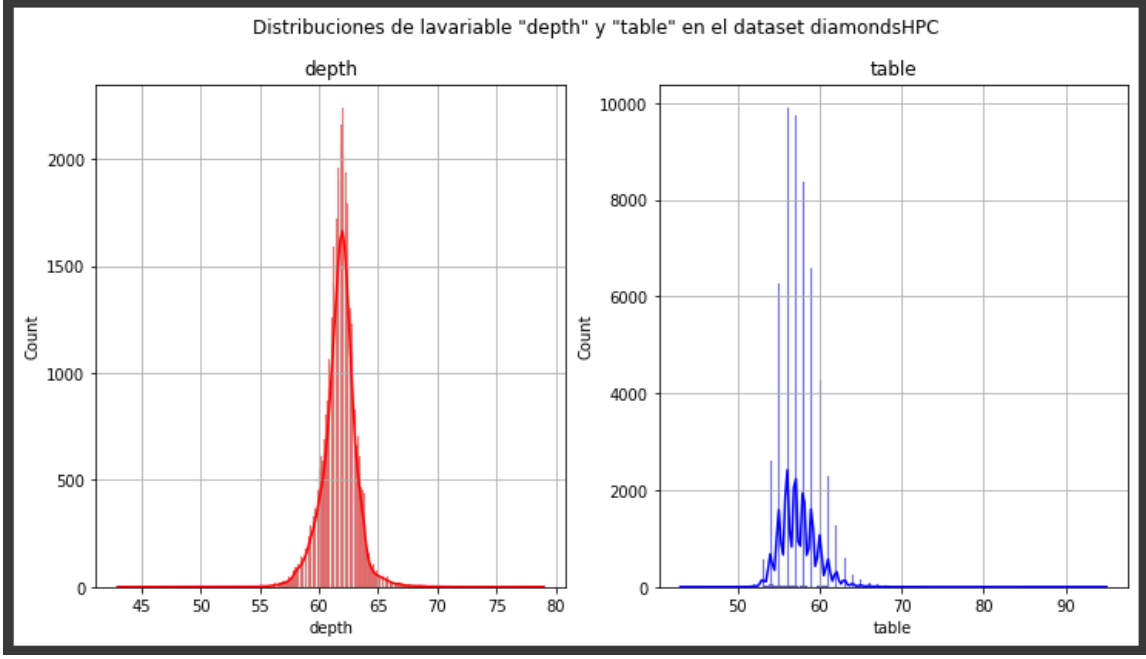
```python
from IPython.core.pylabtools import figsize
# 4.3 -> Se grafica la distribución de la variable independiente 'depth' y 'table'
fig, axes = plt.subplots(1, 2, figsize=(12,6))

fig.suptitle('Distribuciones de lavariable "depth" y "table" en el dataset diamondsHPC')

sns.histplot(df['depth'], ax=axes[0], kde=True, color='r')
axes[0].set_title('depth')
axes[0].grid()

sns.histplot(df['table'], ax=axes[1], kde=True, color='b')
axes[1].set_title('table')
axes[1].grid()

plt.show()
```



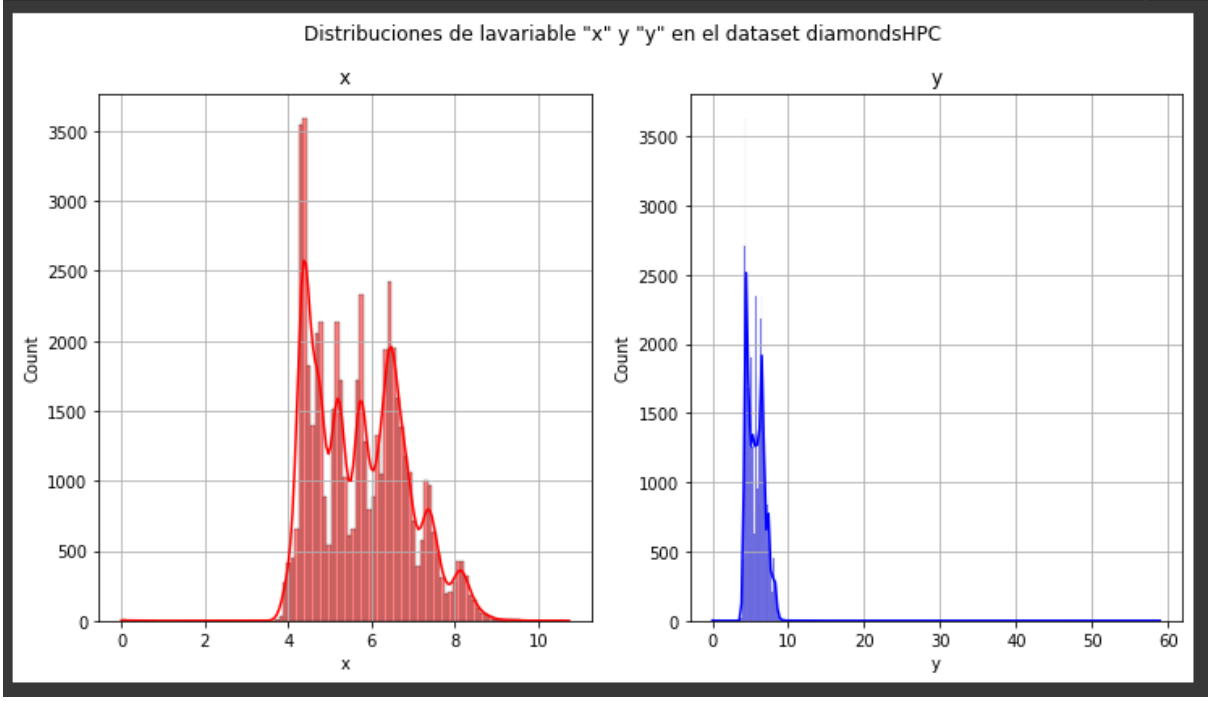
```python
from IPython.core.pylabtools import figsize
# 4.3 -> Se grafica la distribución de la variable independiente 'x' y 'y'
fig, axes = plt.subplots(1, 2, figsize=(12,6))

fig.suptitle('Distribuciones de lavariable "x" y "y" en el dataset diamondsHPC')

sns.histplot(df['x'], ax=axes[0], kde=True, color='r')
axes[0].set_title('x')
axes[0].grid()

sns.histplot(df['y'], ax=axes[1], kde=True, color='b')
axes[1].set_title('y')
axes[1].grid()

plt.show()
```



De las correlaciones anteriormente mostradas, sólo se evidencia que la variable 'depth' sigue una distribución normal.

Como resultado del EDA anteriormente realizado, se espera una baja precisión del modelo sobre este conjunto de datos, esto teniendo en cuenta la correlación que existe entre las variables independientes con la variable dependiente.

Apreciaciones

Según el análisis estadístico realizado, así como las métricas de rendimiento evaluadas, es de esperar un desempeño pobre del modelo en el proceso de predicción.

Teniendo en cuenta la métrica R^2 evaluada sobre los datos de test, el modelo no se recomienda.

Conclusiones

Pese a que no existe una fuerte correlación entre todas las variables independientes y la variable dependiente, no se recomienda utilizar un modelo de Regresión Lineal, esto se puede confirmar con las evidencias anteriormente mostradas, sin embargo, existe cierta correlación de las variables, lo que deja una posibilidad de lograr un desempeño mucho mejor haciendo uso de otras técnicas de Machine Learning.

Referencias

https://www.coursera.org/learn/machine-learning

https://sitiobigdata.com/2018/09/03/como-seleccionar-la-metrica-de-evaluacion-correcta-para-los-modelos-de-aprendizaje-automatico-parte-2-metricas-de-regresion/