ENTREGA FINAL

POR:
Juan David Sandoval
Guerrero Daniela Tuberquia

MATERIA:
Introducción a la inteligencia artificial.

PROFESOR:
Raul Ramos Pollan

UNIVERSIDAD DE ANTIOQUIA FACULTAD DE INGENIERÍA
2022

INDICE

1. INTRODUCCIÓN

La inteligencia artificial está siendo implementada alrededor del mundo, ya que soluciona diferentes problemas que de otra manera sería más tedioso para el ser humano desarrollar dicha actividad. “La inteligencia artificial se puede resumir en tres partes, el estudio de redes neuronales (1950-70), el aprendizaje automático (1980-2010) y actualmente el Deep learning”.

En la industria, la inteligencia artificial se está estableciendo como una ventaja competitiva y también como un valor agregado al producto o servicio final que se está ofreciendo. Esta nueva tecnología está abriendo un nuevo escenario en el desarrollo de las empresas, ya que es una realidad en el ecosistema empresarial, aunque su potencial todavía no es conocido del todo.

Todas las definiciones de IA (inteligencia artificial), llevan a la siguiente idea: Desarrollo de métodos y algoritmos que permitan comportarse como computadores de modo inteligente. Todos los procesos que se llevan a cabo en el cerebro pueden ser analizados, a un nivel de abstracción dado, como procesos computacionales de algún tipo.

En este trabajo se desarrollará la implementación de Machine learning para predecir el precio final de una vivienda la cual cuenta con varias características propias de ellas.

2. PLANTEAMIENTO DEL PROBLEMA.

El modelo que se desarrolla basado en la competencia que tiene como título: “House Price”, el cual dependiendo de las características de una vivienda tales como su superficie, localización, garaje, piscina, año construcción, año de remodelación, entre otros. Se quiere llegar a predecir el valor final de la vivienda en el mercado. El objetivo es realizar un algoritmo de predicción el cual busca tener un error de aproximadamente menor o igual 25%.

2.1 DATASET

El dataset a utilizar puede encontrar en el siguiente link:

https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques/data.

El cual consta de 1460 muestras (casas) y de 80 columnas las cuales describen casi todas las características de las viviendas.

Variable	Descripción	Valores que puede tomarla variable.

MSSubClass	Esta variable identificael tipo de vivienda que se venderá	**20**, vivienda de 1946 y másnuevo todos los estilos. **30**, vivienda 1945 y másantiguo **40**, piso con ático terminado todas lasedades- **45**, 1-½ historia- sin terminar todas las edades.**50**, 1-½, historia terminadade todas las edades. **60**, 2 pisos 1946 y másrecientemente. **70**, 2 pisos 1945 y másantiguos **75**, 2-½ historia todas lasedades. Entre otros valores puedetomar esta variable.
MSZoning	En esta variable describe la clasificaciónde zonificación general de la venta.	**A**, agricultura **C**, comercial **FV**, residencial de puebloflotante **I**, industrial **RH**, residencia de altadensidad. **RL**, residencia de bajadensidad **RP**,parque residencial debaja densidad **RM**, residencial media densidad
Lot Frontage	Pies lineales de calle conectados a lapropiedad.	
Lot Area	Tamaño del lote en piescuadrados.	
Street	Tipo de camino de acceso a la propiedad.	**Grvl,**grava **Pave**,pavimentado
Alley	Tipo de callejón de acceso a la propiedad.	**Grvl,**grava **Pave**,pavimentado **NA**,Sin acceso a callejones
Lot Shape	forma general de la propiedad.	**Reg,** regular. **IR1,** Ligeramente irregular.**IR2,** moderadamente irregular. **IR3,** Irregular.
Land Contour	Planitud de la propiedad	**Lvl,** cerca de Plano/Nivel.**Bnk,** aumento rápido y significativo de grado de calle a edificio. **HLS**,pendiente significativade lado a lado. **Low,** depresión.
Land Contour	Tipos de servicios	**AllPub,** Todos los serviciospúblicos (E, G,

	disponibles	W y S) **NoSewr,** Electricidad, Gas y Agua (Fosa Séptica) **NoSeWa** Solo electricidad y gas **ELO,** Electricidadsolamente
LoadConfig	Configuración de lotes	**Inside,** lote interior **Corner,** lote de esquina **CulDSac,** **Cul-de-sac,FR2,** fachada en dos ladosde la propiedad **FR3,**fachada en tres ladosde la propiedad
Land Slope	Pendientes de la propiedad	**Gtl,**pendiente suave **Mod,**pendiente moderado **Sev,**pendiente grave.
Neighborhood	Ubicaciones físicas dentro de los límites de la ciudad de Ames	**Blmngtn**,BloomingtonHeights **Bluest**e, Bluestem **BrDal**e, Briardale **BrkSide**, Brookside **ClearCr,** Clear Creek **CollgCr**,College Creek **Crawfor,** Crawford **Edwards**,Edwards **Gilbert**,Gilbert **IDOTRR**,Iowa DOT andRail Road
Condition 1:	Proximidad a variascondiciones	**Artery,**Adyacente a la calleprincipal **Feedr,** Adyacente a lacalle alimentadora **Norm,** Normal **RRNn,** Dentro de 200' delFerrocarril Norte-Sur **RRA, adyacente** al Ferrocarril Norte-Sur **PosN,**característica externa positiva cercana: parque, área verde, etc.

2.2 METRICA

La métrica de evaluación principal que se utilizará en este proyecto será el error cuadrático medio (RMSE), el cual consiste es la diferencia entre los valores predichos por un modelo y los valores observados.

$$\sqrt{\frac{1}{N}\sum_{i=1}^{n}(yi - \hat{y}i)^2}$$

Donde N es el número total de datos del dataset, yi es el valor real y $\hat{y}i$ Es el valor de la predicción.

En la métrica de negocio, se desea que las predicciones sean lo suficientemente confiables, para saber el valor de las viviendas y así hacer un análisis más detallado en este negocio inmobiliario.

2.3 VARIABLE OBJETIVO.

Como anteriormente se mencionó la variable objetivo en este Dataset es “Sale Price”, la cual es el precio del inmueble. Se buscará que la predicción sea lo mas cercana posible a la real.

3. EXPLORACION DE LAS VARIABLES.

Para iniciar la exploración de los datos a ser tratados, se usa el archivo “Train”, el cual contiene 81 variables y 1460 muestras, lo primero que se realizar es clasificarla en datos numéricos y datos categóricos, los datos numéricos son subdivididos en: temporales, continuos y discretos, mientras que los datos categóricos son solo de un tipo. A continuación, se muestra el filtrado de los datos mencionados anteriormente:

3.1 FILTRADO DE DATOS NUMERICOS:

Para lograr obtener los datos numéricos, se filtra el dataframe inicial, para este caso se obtienen 38 variables numéricas, las cuales serán subdivididas en las categorías mencionadas anteriormente, los datos numéricos obtenidos son los siguientes:

MSSubClass, LotFrontage, LotArea, OverallQual, OverallCond, YearBuilt, YearRemodAdd, MasVnrArea, BsmtFinSF1, BsmtFinSF2, BsmtUnfSF, TotalBsmtSF, 1stFlrSF, 2ndFlrSF, LowQualFinSF, GrLivArea, BsmtFullBath, BsmtHalfBath, FullBath, HalfBath, BedroomAbvGr, KitchenAbvGr, TotRmsAbvGrd, Fireplaces, GarageYrBlt, GarageCars, GarageArea, WoodDeckSF, OpenPorchSF, EnclosedPorch, 3SsnPorch, ScreenPorch, PoolArea, MiscValMoSold, YrSold, SalePrice.

Luego de obtener los datos numéricos, se filtran en las tres categorías numéricas.

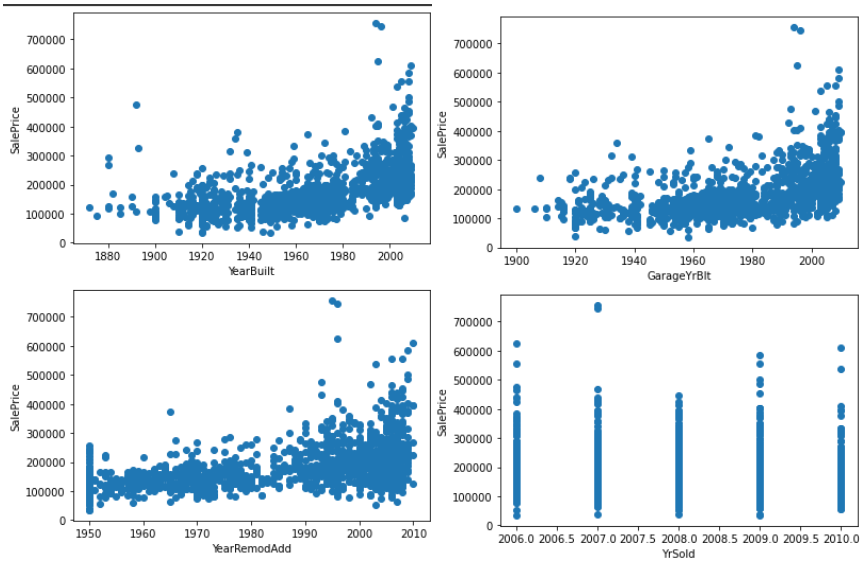
3.1.1 FILTRADO DE DATOS NÚMERICOS TEMPORALES:

En los datos numéricos temporales se obtienen las variables de tipo temporal, los cuales son: año de construcción, año de remodelación, año de remodelación del

garaje y el año de puesta a tierra. A continuación, un resumen de las variables temporales:

	count	mean	std	min	25%	50%	75%	max
YearBuilt	1460.0	1971.267808	30.202904	1872.0	1954.0	1973.0	2000.0	2010.0
YearRemodAdd	1460.0	1984.865753	20.645407	1950.0	1967.0	1994.0	2004.0	2010.0
GarageYrBlt	1379.0	1978.506164	24.689725	1900.0	1961.0	1980.0	2002.0	2010.0
YrSold	1460.0	2007.815753	1.328095	2006.0	2007.0	2008.0	2009.0	2010.0

Gráfica de los datos temporales vs precio de venta:



De las anteriores gráficas se puede concluir que a mayor sea el año de construcción de la vivienda, el año de remodelación o el año de construcción del garaje, mayor es el precio de venta, mientras que para el año de puesta a la venta no se ve ninguna relación especifica, más adelante se busca el nivel de correlación entre estas variables y el precio de venta.

3.1.2 FILTRADO DE DATOS NUMERICOS DISCRETOS.

Para este caso se realizó un filtrado para los datos numéricos de tipo discretos, obteniendo las siguientes 17 variables:

	count	mean	std	min	25%	50%	75%	max
MSSubClass	1460.0	56.897260	42.300571	20.0	20.0	50.0	70.0	190.0
OverallQual	1460.0	6.099315	1.382997	1.0	5.0	6.0	7.0	10.0
OverallCond	1460.0	5.575342	1.112799	1.0	5.0	5.0	6.0	9.0
LowQualFinSF	1460.0	5.844521	48.623081	0.0	0.0	0.0	0.0	572.0
BsmtFullBath	1460.0	0.425342	0.518911	0.0	0.0	0.0	1.0	3.0
BsmtHalfBath	1460.0	0.057534	0.238753	0.0	0.0	0.0	0.0	2.0
FullBath	1460.0	1.565068	0.550916	0.0	1.0	2.0	2.0	3.0
HalfBath	1460.0	0.382877	0.502885	0.0	0.0	0.0	1.0	2.0
BedroomAbvGr	1460.0	2.866438	0.815778	0.0	2.0	3.0	3.0	8.0
KitchenAbvGr	1460.0	1.046575	0.220338	0.0	1.0	1.0	1.0	3.0
TotRmsAbvGrd	1460.0	6.517808	1.625393	2.0	5.0	6.0	7.0	14.0
Fireplaces	1460.0	0.613014	0.644666	0.0	0.0	1.0	1.0	3.0
GarageCars	1460.0	1.767123	0.747315	0.0	1.0	2.0	2.0	4.0
3SsnPorch	1460.0	3.409589	29.317331	0.0	0.0	0.0	0.0	508.0
PoolArea	1460.0	2.758904	40.177307	0.0	0.0	0.0	0.0	738.0
MiscVal	1460.0	43.489041	496.123024	0.0	0.0	0.0	0.0	15500.0
MoSold	1460.0	6.321918	2.703626	1.0	5.0	6.0	8.0	12.0

3.1.3 FILTRADO DE DATOS NUMERICOS CONTINUOS.

Para este caso se realizó un filtrado para los datos numéricos de tipo discretos, obteniendo las siguientes 16 variables:

	count	mean	std	min	25%	50%	75%	max
LotFrontage	1201.0	70.049958	24.284752	21.0	59.00	69.0	80.00	313.0
LotArea	1460.0	10516.828082	9981.264932	1300.0	7553.50	9478.5	11601.50	215245.0
MasVnrArea	1452.0	103.685262	181.066207	0.0	0.00	0.0	166.00	1600.0
BsmtFinSF1	1460.0	443.639726	456.098091	0.0	0.00	383.5	712.25	5644.0
BsmtFinSF2	1460.0	46.549315	161.319273	0.0	0.00	0.0	0.00	1474.0
BsmtUnfSF	1460.0	567.240411	441.866955	0.0	223.00	477.5	808.00	2336.0
TotalBsmtSF	1460.0	1057.429452	438.705324	0.0	795.75	991.5	1298.25	6110.0
1stFlrSF	1460.0	1162.626712	386.587738	334.0	882.00	1087.0	1391.25	4692.0
2ndFlrSF	1460.0	346.992466	436.528436	0.0	0.00	0.0	728.00	2065.0
GrLivArea	1460.0	1515.463699	525.480383	334.0	1129.50	1464.0	1776.75	5642.0
GarageArea	1460.0	472.980137	213.804841	0.0	334.50	480.0	576.00	1418.0
WoodDeckSF	1460.0	94.244521	125.338794	0.0	0.00	0.0	168.00	857.0
OpenPorchSF	1460.0	46.660274	66.256028	0.0	0.00	25.0	68.00	547.0
EnclosedPorch	1460.0	21.954110	61.119149	0.0	0.00	0.0	0.00	552.0
ScreenPorch	1460.0	15.060959	55.757415	0.0	0.00	0.0	0.00	480.0
SalePrice	1460.0	180921.195890	79442.502883	34900.0	129975.00	163000.0	214000.00	755000.0

3.1.4 CORRELACIÓN DE DATOS NUMERICOS.

Después de realizar el filtrado de las variables numéricas y como se había mencionado anteriormente, se procede a realizar un filtrado de correlación con respecto al precio de venta, todo esto haciendo uso del coeficiente de correlación de Pearson, para este caso, se seleccionan las variables con correlación mayor a 0.2, obteniendo las siguientes variables:

LotFrontage,LotArea,OverallQual,YearBuilt,YearRemodAdd,MasVnrArea,BsmtFinSF1,BsmtUnfSF,TotalBsmtSF,1stFlrSF,2ndFlrSF,GrLivArea,BsmtFullBath,FullBath,HalfBath,TotRmsAbvGrd,Fireplaces,GarageYrBlt,GarageCars,GarageArea,WoodDeckSF,OpenPorchSF,SalePrice.

3.2 FILTRADO DE DATOS CATEGORICOS.

Se procede a filtrar los datos de tipo categóricos del dataframe inicial, del cual se obtienen 43 variables de tipo categóricas, dentro de estos datos categóricos, se encuentran variables únicas, que nos indican la categoría a la cual pertenece. En este proceso a cada variable única se le asigna un número dentro de su tipo de dato categórico, como se muestra a continuación:

	MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope
0	3	1	2	3	3	0	4	0
1	3	1	2	3	3	0	2	0
2	3	1	2	0	3	0	4	0
3	3	1	2	0	3	0	0	0
4	3	1	2	0	3	0	2	0
...	...	...	...	...	...	...	...	...
1455	3	1	2	3	3	0	4	0
1456	3	1	2	3	3	0	4	0
1457	3	1	2	3	3	0	4	0
1458	3	1	2	3	3	0	4	0
1459	3	1	2	3	3	0	4	0

1460 rows × 44 columns

3.2.1 CORRELACION DE DATOS CATEGORICOS.

Aplicando para este caso el coeficiente de correlación de Pearson, las variables con una correlación mayor a 0.2 son las siguientes:

Neighborhood,RoofStyle,Foundation,CentralAir,Electrical,PavedDrive,SaleCondition,SalePrice.

4. TRATAMIENTO DE DATOS:

- **Relleno de datos faltantes:**

En el análisis exploratorio, se encontraron con valores faltantes en las diferentes variables que se tienen tanto en los datos Train como en los datos Test. Por lo tanto, se busca estos valores con el siguiente segmento de código:

```
data_missing_cnttes = datatest.isnull().sum()
valores_ftest=data_missing_cnttes[data_missing_cnttes > 0].sort_values(ascending=False)
valores_ftest
```

En el cual nos dirá que columnas tiene datos faltantes y cuantos son de este tipo:

```
TotalBsmtSF    1
GarageCars     1
GarageArea     1
dtype: int64
```

Se opta por rellenar dichos valores con cero como se muestra a continuación:

```
datatest.TotalBsmtSF.fillna(0, inplace=True)
datatest.GarageCars.fillna(int(0), inplace=True)
datatest.GarageArea .fillna(int(0), inplace=True)
```

- **Cambiar todos los datos a int 64:**

Una dificultad que se obtuvo fue que cuando se reemplaza los valore faltantes por cero, el tipo de la columna se convierte en Float 64, por esta razón se busca el tipo que tiene cada columna.

```
datatest.dtypes
```

```
Id               int64
OverallQual      int64
YearBuilt        int64
YearRemodAdd     int64
TotalBsmtSF    float64
1stFlrSF         int64
GrLivArea        int64
FullBath         int64
TotRmsAbvGrd     int64
GarageCars     float64
GarageArea     float64
ExterQual        int64
BsmtQual         int64
KitchenQual      int64
GarageFinish     int64
dtype: object
```

Y se convierte los Float 64 en int 64, la razón de esta conversión es que mas adelante en los modelos no permite este tipo de dato.

```
datatest.astype('int64').dtypes
```

```
Id              int64
OverallQual     int64
YearBuilt       int64
YearRemodAdd    int64
TotalBsmtSF     int64
1stFlrSF        int64
GrLivArea       int64
FullBath        int64
TotRmsAbvGrd    int64
GarageCars      int64
GarageArea      int64
ExterQual       int64
BsmtQual        int64
KitchenQual     int64
GarageFinish    int64
dtype: object
```

5. MODELOS.

5.1 RANDOM FOREST CLASSIFIER.

El bosque de aleatorio o el bosque de decisiones aleatorias es un algoritmo de aprendizaje automático supervisado que se utiliza para la clasificación la regresión y otras tareas mediante arboles de decisión. El clasificador de bosque aleatorio crea un conjunto de árboles de decisión a partir de un subconjunto seleccionado al azar del conjunto de entrenamiento. Es básicamente un conjunto de árboles de decisión de un subconjunto seleccionado al azar del conjunto de entrenamiento y luego recopila los votos de diferentes arboles de decisión para decidir la predicción final.

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_train)
```

```
classifier = RandomForestClassifier(n_estimators=100)
```

```
classifier.fit(X_train, Y_train)
```

El RMSE con este método se obtuvo de:

```
suma=0
for i in range(len(y_test)):
  RMSE=(y_test1[i]-y_pred[i])**2
  suma=suma+RMSE
RMSE1=((1/(len(y_test)))*(suma))**0.5
RMSE1
```

```
67599.12979638366
```

Y el error relativo fue de:

```
np.mean(abs((y_test1-y_pred)/(y_test1)))*100
```

```
27.682543602049638
```

Se miro que variables tienen mas importancia en esta predicción y se encontró lo siguiente:

	Feature	Importance
0	Id	0.123801
6	GrLivArea	0.112367
5	1stFlrSF	0.110251
4	TotalBsmtSF	0.109417
10	GarageArea	0.105548
2	YearBuilt	0.098427
3	YearRemodAdd	0.089644
8	TotRmsAbvGrd	0.059919
1	OverallQual	0.046646
14	GarageFinish	0.036374
12	BsmtQual	0.027776
13	KitchenQual	0.024070
9	GarageCars	0.022069
7	FullBath	0.017688
11	ExterQual	0.016003

Las variables menos importantes como FullBathe y Exterqual solo están aportando el 0.017 y 0.016 respectivamente, este valor es muy pequeño por lo tanto se procede a eliminar y a volver a calcular las predicciones, con esta modificación se obtuvo el siguiente error relativo de:

```
np.mean(abs((y_test1-y_pred1)/(y_test1)))*100
```

```
27.188581644872585
```

Lo que se concluye es que estas variables no son de gran importancia y por esta razón no afecta mucho en el resultado obtenido.

5.2 PIPELINE

El pipeline es el código más común que generará un modelo para cualquier problema de clasificación o de regresión. También generan códigos para entrenamiento y prueba, transforma datos, entre otros.

Con este método no se necesitó cambiar las variables categóricas ya que el mismo método realiza una codificación de estos datos.

```
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OrdinalEncoder
from sklearn.pipeline import Pipeline
from xgboost import XGBRegressor
X_train= datacorrep
Y_train=d["SalePrice"]
ct=ColumnTransformer([('step1',OrdinalEncoder(handle_unknown='use_encoded_value', unknown_value=-1),cat_cols)])
pipeline=Pipeline([('cltf_step',ct), ("Gradient Boost",XGBRegressor(learning_rate=1,random_state=42,n_jobs=5))])
pipeline.fit(X_train,Y_train)
```

```
[16:22:08] WARNING: /workspace/src/objective/regression_obj.cu:152: reg:linear is now deprecated in favor of reg:squarederror.
Pipeline(steps=[('cltf_step',
                 ColumnTransformer(transformers=[('step1',
                                                  OrdinalEncoder(handle_unknown='use_encoded_value',
                                                                 unknown_value=-1),
                                                  array(['Id', 'OverallQual', 'YearBuilt', 'YearRemodAdd', 'TotalBsmtSF',
       '1stFlrSF', 'GrLivArea', 'FullBath', 'TotRmsAbvGrd', 'GarageCars',
       'GarageArea', 'ExterQual', 'BsmtQual', 'KitchenQual',
       'GarageFinish'], dtype=object))])),
                ('Gradient Boost',
                 XGBRegressor(learning_rate=1, n_jobs=5, random_state=42))])
```

El RMSE que se obtuvo con este método fue el siguiente:

```
suma=0
for i in range(len(y_test)):
  RMSE12=(y_test1[i]-Y_pred[i])**2
  suma=suma+RMSE12
RMSE2=(1/(len(y_test1))*(suma))**0.5
RMSE2
```

```
58644.428648184
```

Y el error que se obtuvo es de:

```
np.mean(abs((y_test1-Y_pred)/(y_test1)))*100
```

```
25.116685630030588
```

6. CONCLUSION

- Es necesario tener muy claro los tipos de variables con las que cuenta el dataset y la variable objetivo que se desea pronosticar, con el fin de tener una buena predicción y un error pequeño.
- Teniendo en cuenta las clasificaciones o rangos de las diferentes variables del dataset se puede proceder a hacer el filtrado.
- En el filtrado de las variables hay que tener en cuenta las variables con mayor correlación con la variable objetivo.

7. REFERENCIA

- https://www.kaggle.com/competitions/house-prices-advanced-regression-techniques
- https://www.freecodecamp.org/espanol/news/random-forest-classifier-tutorial-how-to-use-tree-based-algorithms-for-machine-learning/