

```

# Tarea: Predicción del Precio de una Vivienda
# Modelos: Regresión Lineal Múltiple y Red Neuronal Artificial

# Paso 1: Importar bibliotecas necesarias
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam

# Paso 2: Cargar y preparar los datos
# SUBE el archivo 'train.csv' a Colab antes de correr esta celda
data = pd.read_csv('train.csv')

# Ver las primeras columnas para elegir variables numéricas relevantes
print(data.select_dtypes(include=[np.number]).columns.tolist())

# Seleccionamos variables predictoras numéricas
features = ['GrLivArea', 'TotalBsmntSF', 'GarageArea', 'YearBuilt', '1stFlrSF']
target = 'SalePrice'

# Eliminar filas con valores nulos en las columnas seleccionadas
data = data[features + [target]].dropna()

X = data[features].values
y = data[target].values

# Separar en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Paso 3.1: Modelo de Regresión Lineal Múltiple
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)

# Evaluación modelo de regresión
y_pred_lr = lr_model.predict(X_test)
mse_lr = mean_squared_error(y_test, y_pred_lr)
r2_lr = r2_score(y_test, y_pred_lr)
print("Regresión Lineal - MSE:", mse_lr)
print("Regresión Lineal - R2:", r2_lr)

# Paso 3.2: Red Neuronal Artificial (RNA)
nn_model = Sequential()
nn_model.add(Dense(64, input_dim=X_train.shape[1], activation='relu'))
nn_model.add(Dense(32, activation='relu'))
nn_model.add(Dense(1))

# Compilar el modelo
nn_model.compile(optimizer=Adam(learning_rate=0.01), loss='mean_squared_error')

# Entrenar la RNA
history = nn_model.fit(X_train, y_train, epochs=100, batch_size=32, verbose=0, validation_split=0.2)

# Evaluación modelo RNA
y_pred_nn = nn_model.predict(X_test).flatten()
mse_nn = mean_squared_error(y_test, y_pred_nn)
r2_nn = r2_score(y_test, y_pred_nn)
print("Red Neuronal - MSE:", mse_nn)
print("Red Neuronal - R2:", r2_nn)

# Paso 5: Realizar 5 predicciones nuevas
print("\nEjemplos de predicciones (RNA vs Real):")
for i in range(5):
    print(f"Prediccion: {y_pred_nn[i]:.2f}, Real: {y_test[i]:.2f}")

# Paso 6: Comparación general
print("\nResumen de modelos:")
print(f"Regresión Lineal -> MSE: {mse_lr:.2f}, R2: {r2_lr:.2f}")
print(f"Red Neuronal -> MSE: {mse_nn:.2f}, R2: {r2_nn:.2f}")

```

```
↳ ['Id', 'MSSubClass', 'LotFrontage', 'LotArea', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'MasVnrArea', 'BsmtFinSF1', 'B
Regresión Lineal - MSE: 1892748020.9987743
Regresión Lineal - R2: 0.7532376107778491
/usr/local/lib/python3.11/dist-packages/keras/src/layers/core/dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim` argumen
    super().__init__(activity_regularizer=activity_regularizer, **kwargs)
10/10 ----- 0s 8ms/step
Red Neuronal - MSE: 2583822592.0
Red Neuronal - R2: 0.6631404757499695

Ejemplos de predicciones (RNA vs Real):
Prediccion: 119441.20, Real: 154500.00
Prediccion: 273368.34, Real: 325000.00
Prediccion: 124177.23, Real: 115000.00
Prediccion: 167526.77, Real: 159000.00
Prediccion: 241243.58, Real: 315500.00

Resumen de modelos:
Regresión Lineal -> MSE: 1892748021.00, R2: 0.75
Red Neuronal -> MSE: 2583822592.00, R2: 0.66
```