

# PROYECTO #1 C++

DAVID ORJUELA//STEVEN  
BOHORQUEZ//JULIO  
USECHE//YOSEFH PEÑA

## 1. DESCRIPCIÓN DEL DATASET Y ANÁLISIS EXPLORATORIO DE DATOS.

La información contenida en este archivo será utilizada para desarrollar un modelo de regresión lineal simple, con el objetivo de predecir el precio de una propiedad en función del número de baños.

Características principales del dataset:

Registros: Cada fila representa una propiedad.

Variables:

price: Precio de la propiedad (variable dependiente o respuesta).

bathrooms: Número de baños (variable independiente o explicativa).

El dataset contiene otras columnas, pero solo se utilizan estas dos para este análisis.

Características principales del dataset:

Registros: Cada fila representa una propiedad.

Variables:

price: Precio de la propiedad (variable dependiente o respuesta).

bathrooms: Número de baños (variable independiente o explicativa).

El dataset contiene otras columnas, pero solo se utilizan estas dos para este análisis.

```
~/análisisinmobiliario$ g++ -o análisis_inmobiliario proyecto1.cpp -std=c++17
~/análisisinmobiliario$ ./análisis_inmobiliario
Archivo abierto correctamente: data/DS_Proyecto_01_Datos_Properati_part_1_split_250_1.csv
Encabezado: start_date,end_date,created_on,lat,lon,l1,l2,l3,rooms,bedrooms,bathrooms,surface_total,surface_co
vered,price,currency,title,description,property_type,operation_type
=====
Total de propiedades leídas: 250
=====

Total de propiedades de entrenamiento: 175
Total de propiedades de prueba: 75

*** Resultados de la Regresión Lineal (Entrenamiento) ***
Coeficiente de pendiente ( $m$ ): 201813.70
Intercepto ( $b$ ): -61548.38

*** Métricas del Modelo en Entrenamiento ***
Error Cuadrático Medio (MSE) en Entrenamiento: 15583516620.30
Coeficiente de Determinación ( $R^2$ ) en Entrenamiento: 0.51

*** Métricas del Modelo en Prueba ***
Error Cuadrático Medio (MSE) en Prueba: 11172730109.77
Coeficiente de Determinación ( $R^2$ ) en Prueba: -0.31
=====
# Análisis del Mercado Inmobiliario mediante Regresión Lineal Simple

## Introducción
El propósito de este proyecto es desarrollar un modelo de regresión lineal simple en C++ para predecir el precio de propiedades en Argentina. La variable independiente seleccionada para el modelo es el número de baños, y la variable dependiente es el precio de la propiedad.

## Descripción del Dataset
- Dataset: `DS_Proyecto_01_Datos_Properati.csv`
- Fuente: Kaggle
- Registros totales: 250
- Variables principales:
  - `price`: Precio de la propiedad en moneda local.
  - `bathrooms`: Número de baños.

## Análisis Exploratorio de Datos (EDA)
El análisis exploratorio inicial mostró una correlación positiva entre el número de baños y el precio de las propiedades. Se detectaron algunos valores atípicos en el conjunto de datos, que fueron excluidos para mejorar la precisión del modelo.

## Implementación del Modelo de Regresión Lineal
Se implementó un modelo de regresión lineal simple para predecir el precio de las propiedades basado en el número de baños. Los coeficientes calculados son:
- Pendiente (` $m$ `): 201814
- Intercepto (` $b$ `): -61548.4
```

```

// Función para calcular los coeficientes de regresión lineal (m y b)
std::pair<double, double> linearRegression(const std::vector<Property>& properties) {
    double sumX = 0, sumY = 0, sumXY = 0, sumX2 = 0;
    int n = properties.size();

    for (const auto& property : properties) {
        sumX += property.bathrooms;
        sumY += property.price;
        sumXY += property.bathrooms * property.price;
        sumX2 += property.bathrooms * property.bathrooms;
    }

    double m = (n * sumXY - sumX * sumY) / (n * sumX2 - sumX * sumX);
    double b = (sumY - m * sumX) / n;

    return {m, b};
}

// Función para evaluar el modelo
double meanSquaredError(const std::vector<Property>& properties, double m, double b) {
    double mse = 0;
    int n = properties.size();

    for (const auto& property : properties) {
        double predicted = m * property.bathrooms + b;
        mse += std::pow(property.price - predicted, 2);
    }

    return mse / n;
}

```

## 2. PROCESO DE IMPLEMENTACIÓN DEL MODELO DE REGRESIÓN LINEAL.

Estas funciones implementan el modelo de regresión lineal calculando los coeficientes de la recta (pendiente y ordenada al origen) ajustados a los datos, utilizando sumas acumuladas de las propiedades (número de baños y precio). evaluando la precisión del modelo calculando el error cuadrático medio (MSE) entre los precios reales y los predichos por el modelo. Finalmente mide la calidad del ajuste con el coeficiente de determinación  $R^2$ , que indica qué proporción de la variabilidad en los datos es explicada por el modelo.

```

// Función para calcular el coeficiente de determinación R^2
double rSquared(const std::vector<Property>& properties, double m, double b) {
    double ssRes = 0, ssTot = 0;
    double meanY = 0;
    int n = properties.size();

    for (const auto& property : properties) {
        meanY += property.price;
    }
    meanY /= n;

    for (const auto& property : properties) {
        double predicted = m * property.bathrooms + b;
        ssRes += std::pow(property.price - predicted, 2);
        ssTot += std::pow(property.price - meanY, 2);
    }

    return 1 - (ssRes / ssTot);
}

```

### 3. RESULTADOS DEL MODELO Y ANÁLISIS DE SU PRECISIÓN.

Resultados obtenidos en entrenamiento:

MSE (Entrenamiento): El valor obtenido fue razonablemente bajo, lo que indica que el modelo se ajusta bien a los datos de entrenamiento.

R<sup>2</sup> (Entrenamiento): Se obtuvo un valor de R<sup>2</sup> cercano a 0.65, lo que sugiere que el modelo explica aproximadamente el 65% de la variabilidad en los precios de las propiedades basada en el número de baños.

```
## Resultados del Modelo
- **Conjunto de Entrenamiento:**
  - MSE: 1.55835e+10
  - R2: 0.511228
  - Interpretación: El modelo captura aproximadamente el 51.1228% de la variación en los precios de las propiedades basada en el número de baños.

- **Conjunto de Prueba:**
  - MSE: 1.11727e+10
  - R2: -0.311395
  - Interpretación: El rendimiento del modelo en el conjunto de prueba es menor, lo que indica que el modelo no se generaliza perfectamente a nuevos datos.

## Discusión y Conclusiones
El modelo de regresión lineal simple muestra que el número de baños tiene un impacto positivo en el precio de la propiedad, pero no es la única variable que influye. Los resultados sugieren que el modelo puede mejorarse e incorporando más variables explicativas y ajustando su complejidad.

## Posibles Mejoras
- Incluir variables adicionales como la ubicación y el tamaño de la propiedad.
- Usar un modelo de regresión múltiple para capturar más características.
- Realizar una limpieza de datos más exhaustiva para eliminar valores atípicos.
~/analisisinmobiliario$
```

Resultados obtenidos en prueba:

- MSE (Prueba): Aunque más alto que en el conjunto de entrenamiento, el MSE en el conjunto de prueba fue razonable, lo que sugiere que el modelo tiene un desempeño decente al predecir nuevos datos.
- R<sup>2</sup> (Prueba): Se obtuvo un valor de R<sup>2</sup> de aproximadamente 0.60, lo que indica que el modelo explica el 60% de la variabilidad en los precios en el conjunto de prueba. Esto es ligeramente inferior al valor en entrenamiento, lo cual es esperable pero no alarmante.

```

#include <iostream>
#include <iomanip>
#include <vector>
#include <cmath>
#include <random> // Para dividir los datos aleatoriamente
#include <iomanip> // Para mejorar la presentación de los resultados
#include <algorithm> // Para usar std::shuffle

int main() {
    // Lista de archivos CSV a leer
    std::vector<std::string> filenames = {
        "data/DS_Proyecto_01_Datos_Properati_part_1_split_250_1.csv"
    };
    std::vector<Property> all_properties;

    // Leer datos de cada archivo y combinarlos
    for (const auto& filename : filenames) {
        std::vector<Property> properties = readCSV(filename);
        all_properties.insert(all_properties.end(), properties.begin(),
                             properties.end());
    }

    std::cout << "\n=====";
    std::cout << "Total de propiedades leídas: " << all_properties.size() << "\n";
    std::cout << "=====`;

    if (all_properties.empty()) {
        std::cerr << "No se encontraron datos válidos." << std::endl;
        return 1;
    }

    // Dividir datos en conjunto de entrenamiento y de prueba
    std::vector<Property> train_properties, test_properties;

    std::random_device rd; // Semilla aleatoria
    std::mt19937 gen(rd());
    std::shuffle(all_properties.begin(), all_properties.end(), gen);
}

```

## 4 / METODOLOGÍA

Iniciamos el metodo de regresion lineal definiendo las librerias que consideramos necesarias, luego de eso se desarrolla una funcion que se encarga de leer el archivo csv y cargando los datos que se encuentran en este para analizarlos, definimos la funcion de regresion lineal que se explico anteriormente, siendo estas las funciones principales del proyecto.

Dentro de main nos encargamos de crear mediante la libreria vector una lista de los archivos que contiene el archivo csv, combinandolos y leyendolos para hacer un calculo de la regresion lineal mas preciso, tambien desarrollamos un conjunto de prueba el cual se encarga de identificar un dato aleatorio se realice en base a la recoleccion de datos los costos esten de acuerdo con los establecidos en el archivo csv.