PSet #2: Feature Engineering y Modelos Lineales en Airbnb

Curso de Data Mining/Science

24 de febrero de 2025

1. Introducción

En el mercado actual, la industria de la hospitalidad y el turismo ha visto un crecimiento exponencial gracias a plataformas como Airbnb. Estas plataformas ofrecen a los propietarios la oportunidad de obtener ingresos adicionales al arrendar sus propiedades y a los viajeros la posibilidad de encontrar alojamientos de diversa índole y presupuesto. Sin embargo, determinar el precio ideal de un listado continúa siendo un reto importante.

Alinear los intereses de los anfitriones (que buscan maximizar sus ganancias) y de los huéspedes (que desean un precio razonable) exige un análisis cuidadoso de múltiples variables. Como **Data Scientists**, nuestro objetivo será desarrollar un modelo capaz de predecir el precio de cada listado a partir de diversas características disponibles en el conjunto de datos.

2. Contexto de Negocio

Imagina que trabajas como científico de datos en una empresa que asesora a propietarios en la optimización de precios de sus propiedades en alquiler a corto plazo en distintas ciudades, usando plataformas como Airbnb. Tu meta es desarrollar una herramienta analítica que facilite la toma de decisiones sobre cómo fijar el precio de las propiedades de manera competitiva, maximizando ganancias sin perder atractivo en el mercado.

2.1. Problema de Negocio

El problema de negocio se puede resumir de la siguiente forma:

- **Input**: Un conjunto de listados con sus características (*número de habitaciones, ubicación, servicios disponibles, reseñas, calificación de limpieza, etc.*).
- **Output Deseado**: Un modelo que, dadas estas características, prediga el *precio* de un nuevo anuncio con buena precisión.
- **Objetivo**: Encontrar el precio óptimo que permita a los anfitriones mantener un nivel de ocupación alto y, al mismo tiempo, maximizar sus beneficios.

3. Fuente de Datos

Utilizaremos el dataset *Airbnb Price Dataset* que se encuentra en el D2L

4. Objetivos del PSet #2

En esta segunda práctica, profundizaremos en:

1. **Data Wrangling**: Limpieza de datos, tratamiento de valores faltantes, outliers, etc.
2. **Feature Engineering**:
   - Creación de variables derivadas (ej. índices, interacciones, transformaciones).
   - Selección de variables relevantes.
   - Codificación adecuada de variables categóricas.
3. **Modelos Lineales**: Implementación y uso de diversas técnicas de regresión lineal, tanto mediante ecuaciones analíticas como mediante librerías de Python. Se evaluarán los siguientes métodos:
   a) **Regresión Lineal con Ecuación Normal**
      - Implementación propia.
      - Uso de `sklearn` (*LinearRegression*).
   b) **Regresión Lineal con Singular Value Decomposition (SVD)**
      - Implementación con librerías de álgebra lineal.
      - Uso de `sklearn` (*LinearRegression*).
   c) **Regresión Polinomial**
      - Uso de `sklearn` (`PolynomialFeatures` + `LinearRegression`).
      - Análisis del grado del polinomio y su efecto en el sobreajuste.
   d) **Regresión Lineal con Batch Gradient Descent (BGD)**
      - Implementación propia de BGD.
      - Uso de `sklearn` con `SGDRegressor` (ajustado a modo batch).
   e) **Regresión Lineal con Stochastic Gradient Descent (SGD)**
      - Implementación propia de SGD.
      - Uso de `sklearn` (`SGDRegressor`).
   f) **Lasso Regression** (sólo librería)
   g) **Ridge Regression** (sólo librería)
4. **Model Training & Evaluation**:
   - Separar datos en entrenamiento y prueba (y opcionalmente en validación).
   - Métricas de error: MSE, RMSE, MAE, R^2.
   - Comparar y analizar performance de los distintos modelos.

5. Pasos Detallados

5.1. 1. Exploración y Limpieza de Datos (Data Wrangling)

a) Cargar el dataset y examinar la estructura y el tipo de datos.

b) Detectar y manejar valores faltantes.

c) Identificar y tratar outliers relevantes para la variable *precio*.

d) Analizar y eliminar columnas irrelevantes o redundantes (ej. columnas ID, nombres muy específicos, etc.).

5.2. 2. Feature Engineering

a) Crear variables derivadas (por ejemplo, *total_amenities*, *ratios* o *categorías*).

b) Usar codificación adecuada para variables categóricas (One-Hot Encoding, Label Encoding, etc.).

c) Escalado de variables numéricas si se considera necesario (Normalización/Standardización).

d) Seleccionar variables con mayor correlación o importancia para la tarea de modelado.

5.3. 3. Modelado y Evaluación

Modelo a entrenar: *Predicción del Precio de la Propiedad*

a) Separar el dataset en conjunto de entrenamiento y prueba. Opcionalmente, dividir un conjunto de validación o usar validación cruzada.

b) Entrenar y evaluar cada uno de los modelos lineales listados:

   i) Regresión Lineal con Ecuación Normal (implementación propia + `LinearRegression` de `sklearn`).

   ii) Regresión Lineal con SVD (implementación propia + `LinearRegression` de `sklearn`)).

   iii) Regresión Polinomial (`PolynomialFeatures` + `LinearRegression`).

   iv) Regresión Lineal con Batch Gradient Descent (implementación propia + `SGDRegressor` en modo batch).

   v) Regresión Lineal con Stochastic Gradient Descent (implementación propia + `SGDRegressor`).

   vi) Lasso Regression (sólo uso de `Lasso` en `sklearn`).

   vii) Ridge Regression (sólo uso de `Ridge` en `sklearn`).

c) Evaluar los modelos usando métricas de error (RMSE, MAE, R^2) y comparar resultados.

d) Realizar un análisis sobre cuál modelo ofrece la mejor solución al problema de negocio.

6. Entrega y Estructura de Carpeta en GitHub

Se solicita que cada equipo o estudiante suba un repositorio a GitHub con la siguiente estructura de carpetas (**recomendado** para mantener buenas prácticas en proyectos de Data Science):

```
.
 data
    raw
    processed
 notebooks
    1_exploratory_data_analysis.ipynb
    2_feature_engineering.ipynb
    3_model_training.ipynb
    4_evaluation_and_results.ipynb
 src
    data_wrangling.py
    feature_engineering.py
    models.py
 models
    (archivos .pkl o .h5, etc. para guardar modelos entrenados)
 README.md
 requirements.txt
```

- **data/raw**: Aquí colocarán el dataset original descargado de D2L.
- **data/processed**: Conjunto de datos procesados (después de limpiezas y transformaciones).
- **notebooks**: Notebooks de Jupyter en orden cronológico que muestren el flujo de trabajo:
  - `1_exploratory_data_analysis.ipynb`: Análisis exploratorio y visualización de datos.
  - `2_feature_engineering.ipynb`: Creación de nuevas features y selección de variables.
  - `3_model_training.ipynb`: Implementación y entrenamiento de los distintos modelos lineales.
  - `4_evaluation_and_results.ipynb`: Evaluación de métricas, comparativas y conclusiones.
- **src**: Scripts de Python con funciones auxiliares (ejemplo: funciones de limpieza, ingeniería de características, clases de modelos, etc.).
- **models**: Guardar aquí los modelos entrenados, pesos, checkpoints, etc.
- **README.md**: Explicación general del proyecto, cómo ejecutarlo y dependencias.

- **requirements.txt**: Lista de librerías necesarias (`numpy`, `pandas`, `sklearn`, `matplotlib`, `seaborn`, etc.).

7. Criterios de Evaluación

1. **Calidad de la Limpieza de Datos** (manejo de valores faltantes, outliers, etc.).
2. **Aplicación Correcta de Feature Engineering** (nuevas columnas, codificaciones, etc.).
3. **Implementación de los Modelos** (tanto en su versión propia como con librerías).
4. **Manejo de las Métricas y Análisis** (interpretación de resultados y métricas).
5. **Estructura de Carpeta y Documentación** (organización del repositorio y claridad en la explicación).
6. **Conclusiones de Negocio** (cómo el modelo soluciona el problema propuesto).

8. Conclusiones

Este PSet #2 está diseñado para reforzar los conocimientos en regresión lineal y sus variantes, además de poner en práctica el proceso completo de un proyecto de ciencia de datos: desde la exploración y limpieza de datos hasta la entrega y comparación de modelos. Al finalizar, serás capaz de:

- Identificar y aplicar las mejores prácticas de limpieza y manipulación de datos.
- Construir y evaluar modelos lineales de manera fundamentada, comprendiendo la utilidad de cada tipo de modelo en diferentes escenarios.
- Entregar un proyecto de Data Science bien estructurado, escalable y con fácil trazabilidad.