Predicción de precios de viviendas mediante Redes Neuronales Profundas

Daniel Erick Sanchez Trujillo
Facultad de Ciencias Puras y Naturales
Universidad Mayor de San Andrés
La Paz, Bolivia
daniel.sanchez@agetic.gob.bo

Felipe Roberto Sanchez Saravia
Facultad de Ciencias Puras y Naturales
Universidad Mayor de San Andrés
La Paz, Bolivia
sanchezsaravia@gmail.com

Sergio Alejandro Paucara Saca
Facultad de Ciencias Puras y Naturales
Universidad Mayor de San Andrés
La Paz, Bolivia
sergiopaucara@gmail.com

***Abstract*—Este trabajo aborda la predicción de precios de viviendas mediante la aplicación de Redes Neuronales Profundas (DNN) sobre el conjunto de datos *House Prices* de Kaggle. Se implementó un flujo de trabajo integral que abarca desde el análisis exploratorio de datos hasta el entrenamiento de un modelo de regresión, haciendo uso de Google Colab como entorno de ejecución. El proceso incluyó preprocesamiento robusto, reducción de variables irrelevantes, y aplicación de técnicas de regularización como Batch Normalization y Dropout. Se discute la arquitectura del modelo, el impacto de las decisiones de diseño sobre el rendimiento, y se plantean propuestas para futuras mejoras.**

***Index Terms*—Deep Learning, Regresión, DNN, Kaggle, Preprocesamiento, House Prices.**

I. Introducción

La estimación del precio de una vivienda a partir de sus características físicas y de entorno es un problema clásico en ciencia de datos, con aplicaciones en la valuación de inmuebles, otorgamiento de créditos hipotecarios y análisis de inversión. Los modelos predictivos permiten automatizar este proceso, minimizando la influencia de sesgos subjetivos. En este estudio, se aborda la predicción de precios utilizando una Red Neuronal Profunda (DNN), entrenada sobre el dataset *House Prices: Advanced Regression Techniques* de Kaggle [1].

II. Estado del arte

Históricamente, las técnicas más empleadas para este tipo de problemas han sido la regresión lineal, *Lasso* y *Ridge Regression* debido a su simplicidad y capacidad de interpretación. Modelos basados en *ensembles* como *Random Forests* y *Gradient Boosting Machines (GBM)* han demostrado ser altamente efectivos, aprovechando interacciones no lineales entre variables. Más recientemente, las Redes Neuronales Profundas han emergido como una alternativa viable al capturar relaciones complejas en grandes volúmenes de datos, siempre que se disponga de un preprocesamiento adecuado y técnicas de regularización eficientes.

III. Área de aplicación

La predicción de precios de vivienda impacta directamente en:

- Procesos de valuación y tasación inmobiliaria.
- Análisis de riesgo para entidades financieras.
- Estimaciones de mercado para decisiones de inversión.
- Automatización de plataformas de compraventa de bienes raíces.

IV. Metodología

A. Dataset

El dataset utilizado pertenece a la competencia de Kaggle *House Prices: Advanced Regression Techniques* [1]. Consta de 1460 registros en el conjunto de entrenamiento y 1459 en el conjunto de prueba, con 81 y 80 variables respectivamente, donde `SalePrice` es la variable objetivo.

B. Análisis Exploratorio

Se realizó una inspección inicial de la estructura de datos, identificando variables con alta cantidad de valores nulos, distribuciones asimétricas en atributos numéricos (especialmente `SalePrice`), y relaciones de alta correlación entre variables como `GrLivArea` y `TotalBsmtSF`.

C. Reducción de Variables

Se eliminaron atributos con más del 30% de valores nulos y aquellas con baja correlación (coeficiente ¡ 0.1) respecto a `SalePrice`. Además, se removieron variables redundantes que presentaban alta multicolinealidad.

D. Limpieza y Tratamiento de Datos Faltantes

Como parte del proceso de limpieza, se realizó un análisis exhaustivo de los valores nulos en el conjunto de datos. Se utilizó un mapa de calor para visualizar la distribución de valores faltantes, mostrado en la Fig. 1. Este análisis permitió identificar variables como `GarageCars` y `BsmtUnfSF` con registros incompletos, los cuales fueron imputados utilizando la mediana de cada atributo, asegurando coherencia con la naturaleza de las variables numéricas.

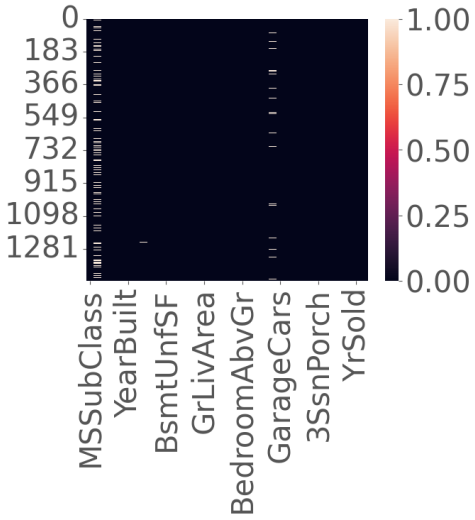

Fig. 1. Mapa de calor de datos faltantes en el dataset de entrenamiento.

Asimismo, se evaluó la distribución de la variable objetivo `SalePrice` mediante un histograma (Fig. 2), observando una clara asimetría positiva, lo que sugiere la pertinencia de considerar transformaciones logarítmicas en futuras iteraciones del modelo para mejorar la linealidad de la relación entre las variables independientes y el objetivo.

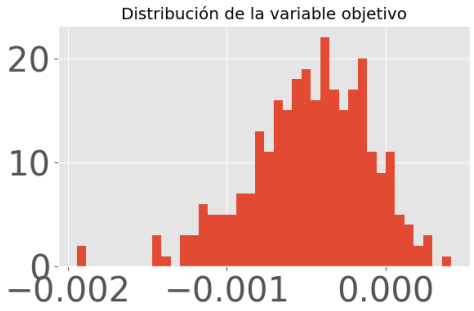

Fig. 2. Distribución de la variable objetivo `SalePrice`.

E. Preprocesamiento

Se implementó un pipeline de preprocesamiento utilizando `ColumnTransformer`, con las siguientes etapas:

- Imputación de valores nulos (mediana para variables numéricas, moda para categóricas).
- Codificación One-Hot para atributos categóricos.
- Normalización de las variables numéricas mediante MinMaxScaler.

F. Arquitectura del Modelo

La Red Neuronal Profunda (DNN) se diseñó con una estructura secuencial de capas densas, incorporando técnicas de regularización en las primeras etapas. La Tabla I detalla la configuración de cada capa, incluyendo el tipo de activación y las técnicas aplicadas.

TABLE I
ARQUITECTURA DE LA RED NEURONAL PROFUNDA (DNN)

Capa	Neuronas	Activación	Regularización
Densa 1	256	ReLU	BatchNorm + Dropout(0.2)
Densa 2	128	ReLU	BatchNorm + Dropout(0.2)
Densa 3	64	ReLU	-
Salida	1	Lineal	-

G. Entrenamiento

El modelo fue entrenado utilizando el optimizador Adam (learning rate = 1e-3), con función de pérdida MSE y métricas MAE/RMSE. Se aplicó EarlyStopping (paciencia 20 épocas) y ReduceLROnPlateau (factor 0.5, paciencia 5) para dinamismo en el ajuste de la tasa de aprendizaje. El entrenamiento se realizó con `batch_size = 256` durante un máximo de 200 épocas.

H. Análisis del Proceso de Entrenamiento

Durante el entrenamiento del modelo se registró la evolución de la función de pérdida (`loss`) para el conjunto de entrenamiento y validación. En la Fig. 3 se observa cómo la pérdida de entrenamiento disminuye de manera pronunciada en las primeras épocas, estabilizándose conforme el modelo converge.

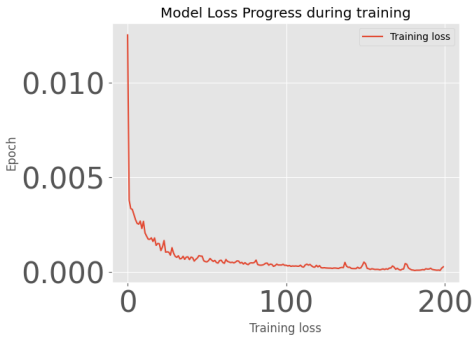

Fig. 3. Progreso de la función de pérdida (Training Loss) durante el entrenamiento.

La Fig. 4 muestra la comparación entre la pérdida de entrenamiento y la de validación. Se aprecia que, si bien la validación presenta mayores oscilaciones debido a la menor cantidad de datos, no se observa un incremento significativo que indique sobreajuste, lo cual valida la efectividad de las técnicas de regularización aplicadas.

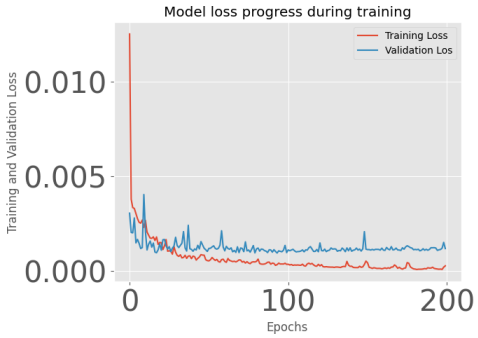

Fig. 4. Comparación entre la función de pérdida de entrenamiento y validación.

V. Resultados

La evaluación del modelo se realizó sobre un conjunto de validación hold-out (20% del dataset de entrenamiento). Las métricas de desempeño obtenidas fueron:

- MAE: 0.0291
- MSE: 0.0022
- RMSE: 0.0470
- R^2: 0.8506
- R^2 Ajustado: 0.8295

El análisis de residuales indicó una distribución aleatoria, evidenciando que el modelo no presenta sesgos sistemáticos importantes en la muestra de validación.

A. Análisis de Predicción vs Valor Real

Para evaluar la precisión de las predicciones del modelo, se construyó un gráfico de dispersión comparando las predicciones normalizadas contra los valores reales de `SalePrice`. En la Fig. 5 se observa que la mayoría de las predicciones se alinean a lo largo de la línea ideal (y = x), lo que indica un buen ajuste. Sin embargo, se identifican algunos casos donde el modelo subestima o sobreestima ligeramente los precios, situación esperable en problemas de alta variabilidad como el inmobiliario.

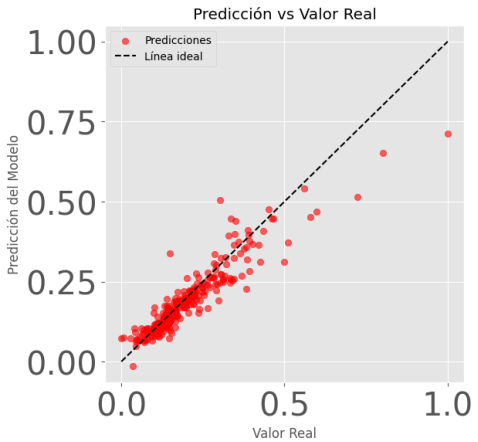

Fig. 5. Comparación de las predicciones del modelo frente a los valores reales.

B. Comparación con Modelo Random Forest

Como parte del análisis comparativo, se evaluó un modelo adicional basado en Random Forest (Regresión), entrenado bajo las mismas condiciones de preprocesamiento. Los resultados obtenidos fueron los siguientes:

- **Random Forest (Regresión):**
  - MAE: 0.025
  - MSE: 0.002
  - RMSE: 0.040
  - R^2: 0.890

En la Tabla II se presenta la comparación directa entre el modelo DNN propuesto y el modelo Random Forest. Si bien el Random Forest presenta un desempeño levemente superior en términos de MAE y RMSE, la diferencia es marginal, destacando la capacidad de la DNN de aproximarse a estos resultados sin un ajuste exhaustivo de hiperparámetros.

TABLE II
Comparación de Desempeño entre DNN y Random Forest

Métrica	DNN	Random Forest
MAE	0.0291	0.0250
RMSE	0.0470	0.0400
R^2	0.8506	0.8900

Este resultado valida la robustez de la arquitectura DNN propuesta, siendo una alternativa viable a modelos tradicionales basados en árboles, especialmente considerando su capacidad de escalar y generalizar ante datasets más complejos.

VI. Discusión

Los resultados demuestran la viabilidad de utilizar una DNN para la predicción de precios inmobiliarios, siempre que se aplique un preprocesamiento adecuado. Si bien el desempeño es competitivo frente a modelos tradicionales, se observó que la arquitectura puede beneficiarse de mejoras adicionales como:

- Aplicación de transformaciones logarítmicas sobre `SalePrice` y variables asimétricas.
- Regularización L2 en las capas densas.
- Validación cruzada K-Fold para una estimación más robusta.
- Optimización de hiperparámetros mediante Grid Search o Bayesian Optimization.

VII. Conclusiones

Se implementó un modelo de regresión basado en Redes Neuronales Profundas para la predicción de precios de viviendas, siguiendo un flujo metodológico reproducible en Google Colab. La reducción de variables y la aplicación de técnicas de regularización resultaron fundamentales para mitigar el sobreajuste. Como trabajo futuro se propone profundizar en la ingeniería de variables y explorar arquitecturas más complejas bajo esquemas de validación cruzada.

REFERENCES

[1] Kaggle, “House Prices: Advanced Regression Techniques.” Disponible: https://www.kaggle.com/c/house-prices-advanced-regression-techniques

[2] M. Abadi et al., “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,” 2015.