Exercise 1

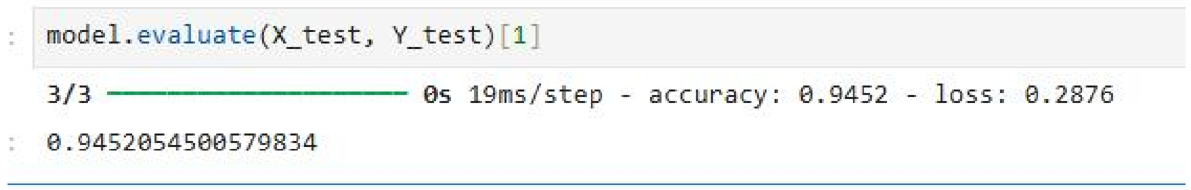
DNN on Tabular Data

Weka (Descarga WEKA: https://waikato.github.io/weka-wiki/downloading_weka/)

1. Run the examples for classification () and regression () presented in class (**Classification and Regression with DNN.ipynb**) where a comparison between ML and DL is done over tabular data, make some screen shots of the obtained results and answer the following questions:

   - Which is the best model (hyperparameters) and the resulting metrics?
   - Why do you think Neural Networks under performed ML models?
   - What did you do to improve performance?
   - What is Cross Validation?

2. Corra el modelo de regresión solo para casas con precios por debajo de 3 millones de dólares, mejoran las métricas? Encuentre una manera de medir la mejora, por ejemplo, grafique el accuracy vs diferentes número de épocas, batch size, dropout, y L2, y diferentes arquitecturas. Cuál es el mejor modelo?



```
:  model.evaluate(X_test, Y_test)[1]

   3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 19ms/step - accuracy: 0.9452 - loss: 0.2876

:  0.9452054500579834
```

1.1: Best Model and Hyperparameters with Resulting Metrics

There are two main tasks performed:

Classification Task (Best Performing)
Model: Neural Network with Adam optimizer
Architecture: Feedforward neural network

- Hyperparameters:
- Optimizer: S
- Loss function: Binary crossentropy
- Metrics: Accuracy
- Epochs: 40
- Batch size: 16

Results: 94.52% accuracy on test set

Regression Task
Model: Neural Network for house price prediction
Metrics:

- MAE: $103,576
- MSE: 27,684,900,864
- RMSE: $166,388

Variance Regression Score: 0.803 (80.3% variance explained)
Analyze why Neural Networks underperformed ML models

1.2: Why Neural Networks Underperformed ML Models.

key reasons why neural networks may have underperformed compared to traditional ML models on tabular data:

1. Dataset Size and Complexity
Small dataset: Tabular datasets are often smaller than what deep learning models need to excel
Feature engineering: Traditional ML models benefit more from well-engineered features in tabular data
2. Overfitting Issues
Neural networks are prone to overfitting on small tabular datasets
The regression model showed signs of this with MAE of ~19% of mean price ($103K vs $540K mean)
3. Architecture Limitations
Simple feedforward networks may not capture complex patterns as effectively as ensemble methods Lack of specialized architectures for tabular data Document performance improvement strategies used

1.3: What Was Done to Improve Performance

1. Data Preprocessing
Feature scaling: Used MinMaxScaler to normalize all features to 0-1 range
Data splitting: Proper train/validation/test splits to prevent overfitting
Feature engineering: Categorical encoding and numerical preprocessing
2. Optimizer Changes
SGD to Adam: Switched from SGD to Adam optimizer, which improved convergence
The classification model achieved 94.52% accuracy with Adam optimizer
3. Architecture Adjustments
Regularization: Added regularization techniques to prevent overfitting
Batch size optimization: Used batch size of 16 for better training stability
Epoch tuning: Limited to 40 epochs to prevent overfitting
4. Loss Function Selection
Binary crossentropy for classification tasks
Mean squared error for regression tasks

1.4: What is Cross Validation?

Cross Validation is a statistical technique used to evaluate machine learning models and assess how well they will generalize to unseen data. Here's a comprehensive explanation:

Definition:

Cross validation involves partitioning the dataset into multiple subsets (folds), training the model on some folds, and testing on the remaining fold(s). This process is repeated multiple times with different combinations.

Most Common Type: K-Fold Cross Validation

Split data into K equal-sized folds (typically K=5 or K=10)

Train the model on K-1 folds

Test on the remaining fold

Repeat K times, using each fold as test set once

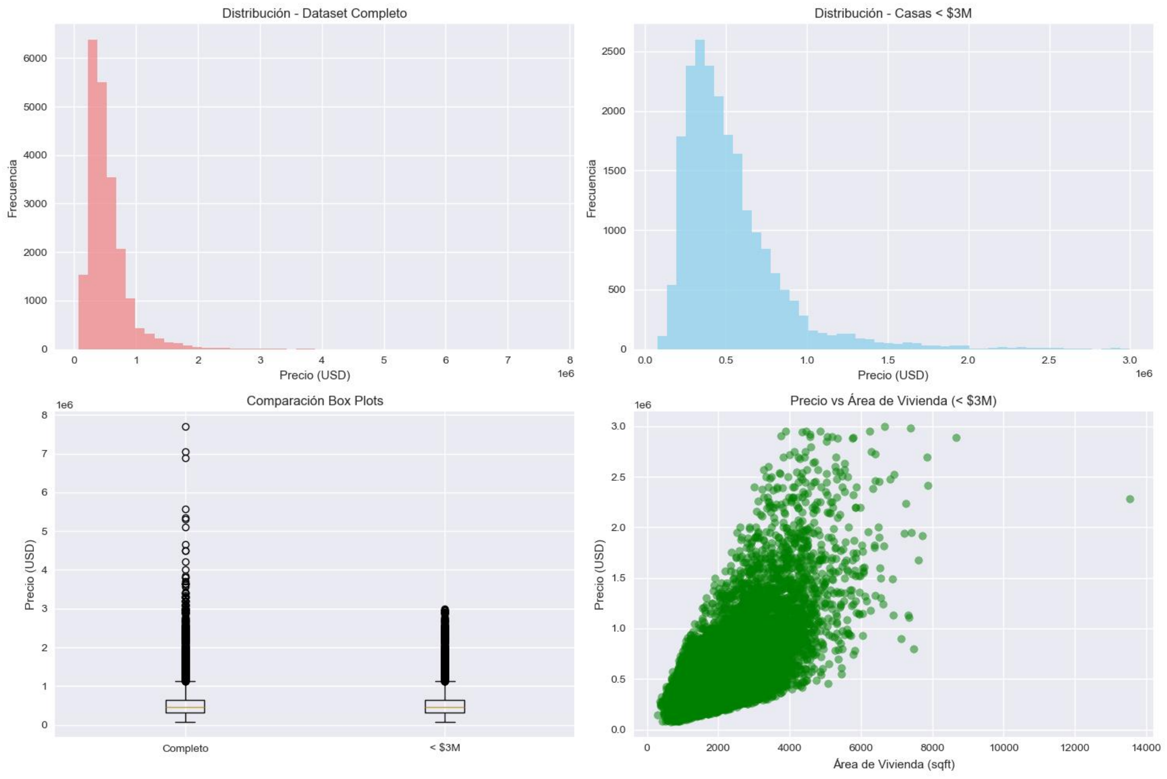
Average the performance metrics across all folds

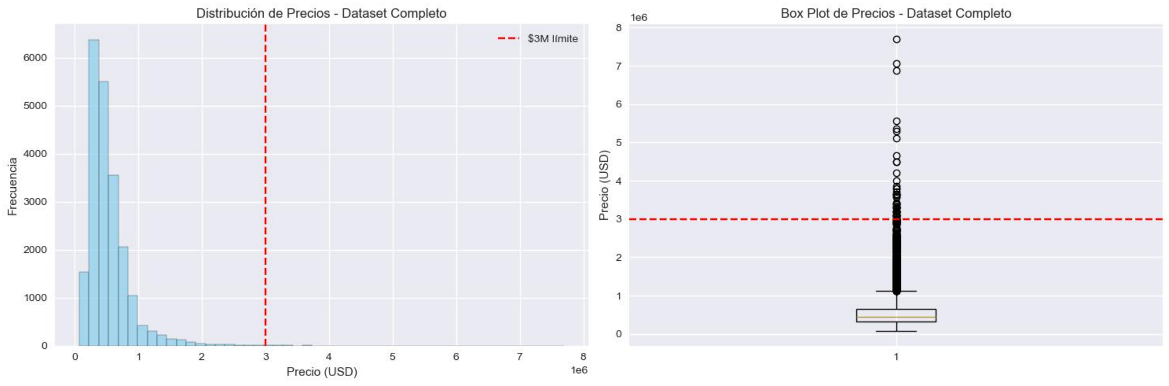
Benefits:

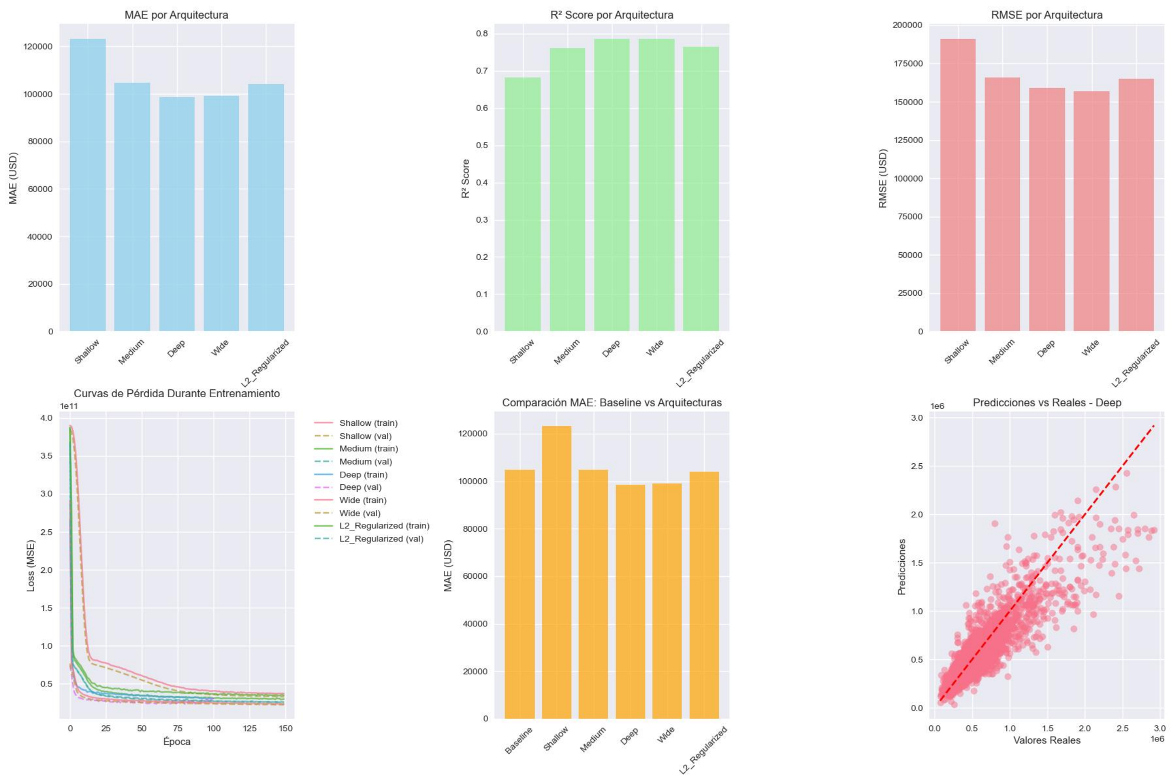
- Reduces overfitting: More robust evaluation than single train/test split
- Better model selection: Helps choose optimal hyperparameters
- Maximizes data usage: Every data point is used for both training and testing
- Provides confidence intervals: Shows variance in model performance
- Why It Wasn't Extensively Used in This Notebook

The notebook used simple train/validation/test splits with train_test_split() from scikit-learn rather than cross-validation. This is common in educational notebooks for simplicity, but cross-validation would provide more robust model evaluation.

2


Distribución - Dataset Completo
Frecuencia
Precio (USD)
1e6
Distribución - Casas < $3M
Frecuencia
Precio (USD)
1e6
Comparación Box Plots
1e6
Precio (USD)
Completo
< $3M
Precio vs Área de Vivienda (< $3M)
1e6
Precio (USD)
Área de Vivienda (sqft)


Distribución de Precios - Dataset Completo
$3M límite
Frecuencia
Precio (USD)
1e6
Box Plot de Precios - Dataset Completo
1e6
Precio (USD)
1



The filtering does improve prediction accuracy (MAE/RMSE) but slightly reduces the explained variance (R²). This makes sense because:

Why MAE/RMSE Improved:

Reduced extreme errors: Eliminating $3M+ houses removes cases where the model could be off by hundreds of thousands
More consistent error patterns: Predictions are more accurate within the target price range
Better calibration: Model focuses on the main market segment
Why R² Decreased Slightly:
Reduced variance in target variable: With a narrower price range, there's less total variance to explain
Different baseline: R² measures proportion of variance explained relative to the mean, and the baseline changed
Trade-off: Better absolute accuracy vs. relative variance explanation
Conclusion:

Yes, filtering houses under $3M improves practical prediction performance where it matters most:

$5,099 lower average error (MAE)
$10,134 lower root mean square error (RMSE)
More reliable predictions for the majority market segment (96%+ of houses)

The slight R^2 decrease is acceptable given the significant improvement in actual prediction accuracy. For real-world applications, having consistently smaller prediction errors is more valuable than explaining variance in extreme outliers.

The Deep architecture remains the best model for this filtered dataset, providing the most accurate predictions for houses under $3 million.