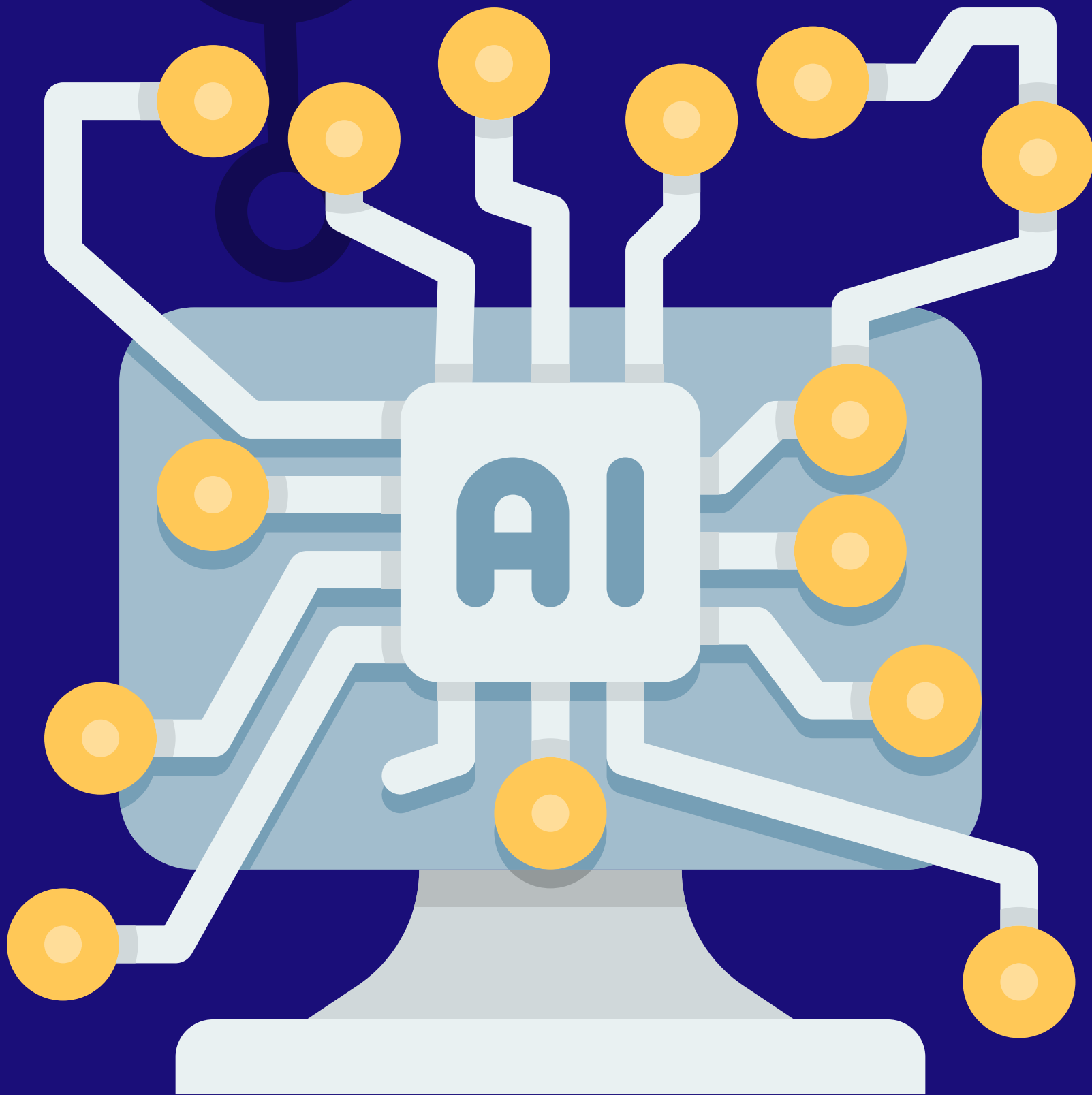


Implementing Linear Regression with AI for Salary Prediction

Deep Learning Project

Jose Luis Pineda Barrera
Daniela Fuentes Bello



Proyect Description

Theme: Implementation of a regression model to estimate the basic salary of a public servant in Colombia.

Objective:

- Use a previously cleaned and adjusted dataset for numerical data.
- Select the input features and the Basic Salary Allowance as the target variable.
- Split the dataset into training and test batch's(80/20).

Activities:

- Implement a regression model to estimate the Basic Salary Allocation.
- Manually tune the model hyperparameters, such as loss function, initializermethod, optimizer, learning rate, number of epochs, batch size, to minimize the loss.

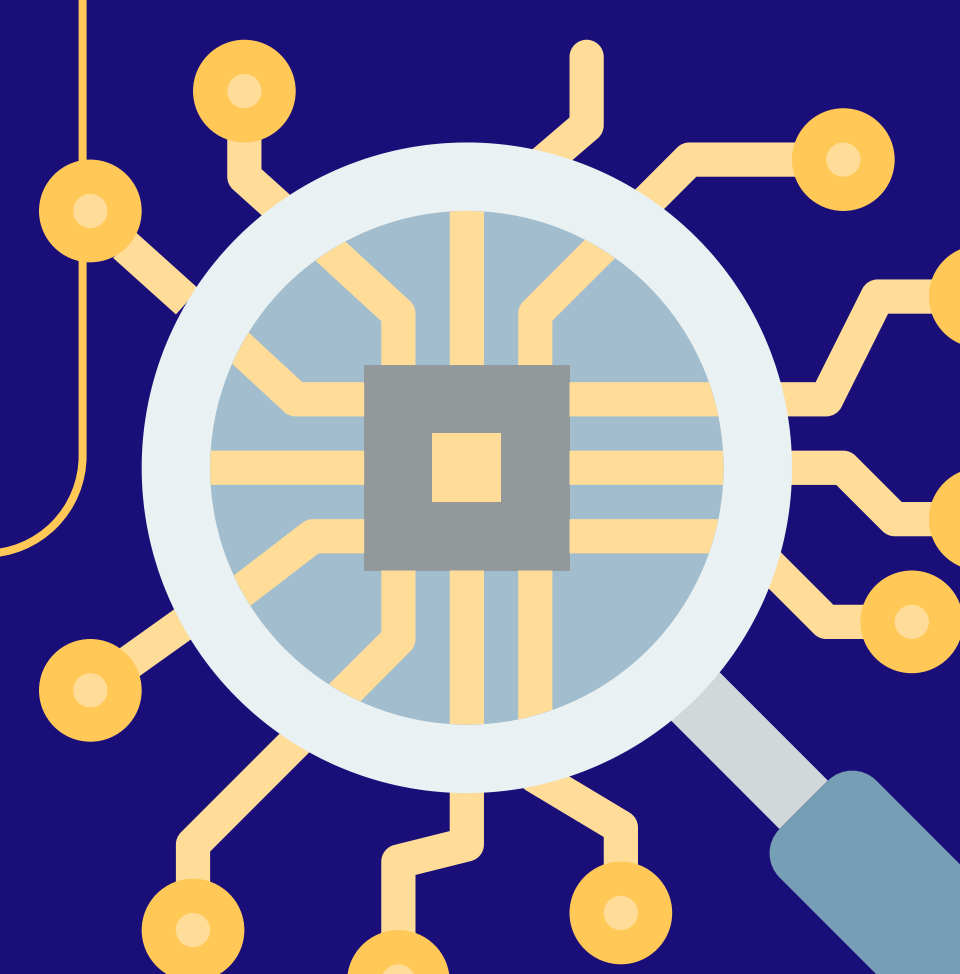


DATA LOADING AND PREPARATION

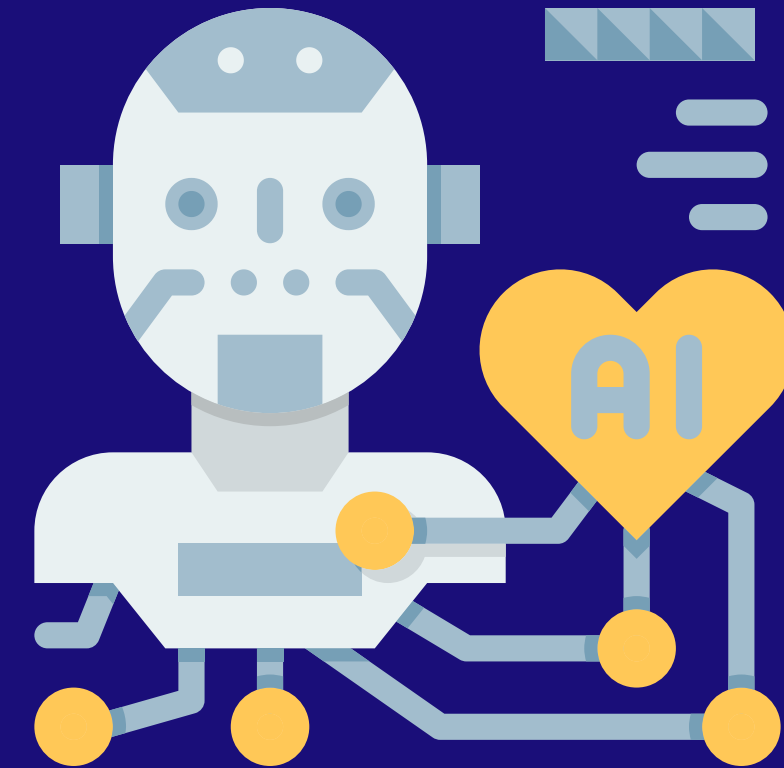
Loading Dataset:

- The dataset is loaded from a CSV file using the pandas library.
- An initial cleanup is performed by removing rows with missing values (dropna).

Preparation:

- Columns irrelevant to the analysis, such as names, IDs, and specific locations, are removed.
 - Numeric columns with commas as decimal separators are converted to float for processing.
- 

SELECTION OF FEATURES



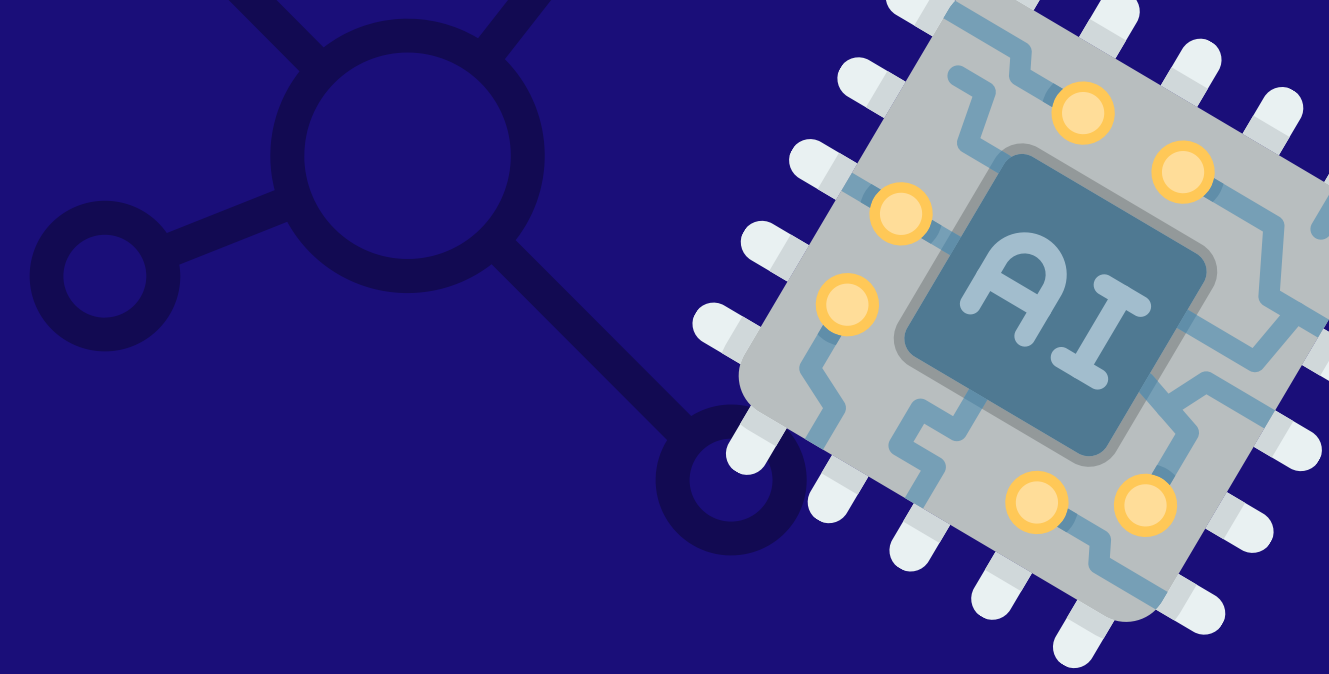
Entry characteristics (X):

- Educational Level
- Hierarchical Level of Employment
- Months of Public Experience
- Months of Private Experience
- Months of Teaching Experience
- Dependence on Current Employment

Target variable (y):

- Basic Salary Allocation

DATASET SPLIT



- The dataset is split into training (80%) and test (20%) sets using sklearn's `train_test_split`.
- The split ensures that the model is evaluated on data not seen during training.

80%

TRAIN

20%

TEST

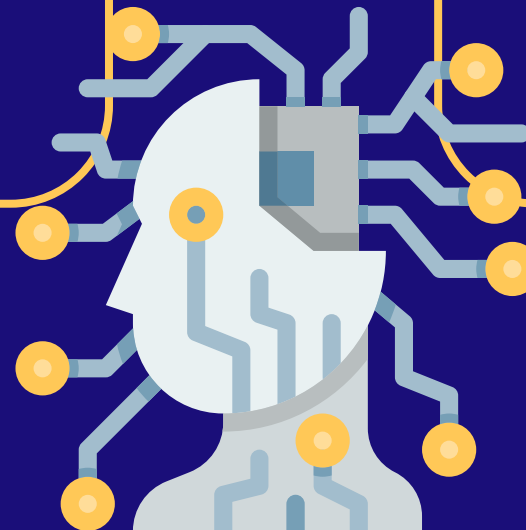
MODEL CONSTRUCTION

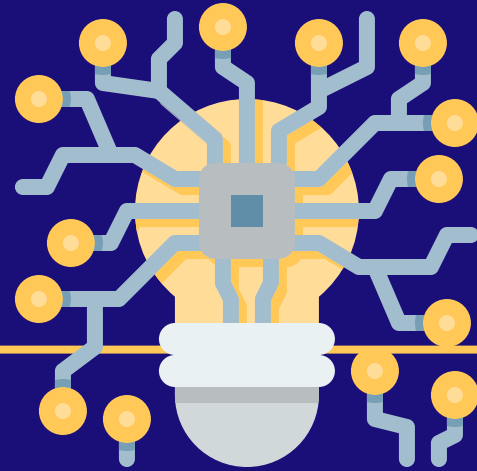
NEURAL NETWORK ARCHITECTURE

- Sequential model with three dense layers:
- Layer 1: 256 neurons, ReLU activation
- Layer 2: 64 neurons, ReLU activation
- Layer 3: 32 neurons, ReLU activation
- Output layer: 1 neuron for regression prediction

OPTIMIZATION

- Adam optimizer with a learning rate of 0.1.



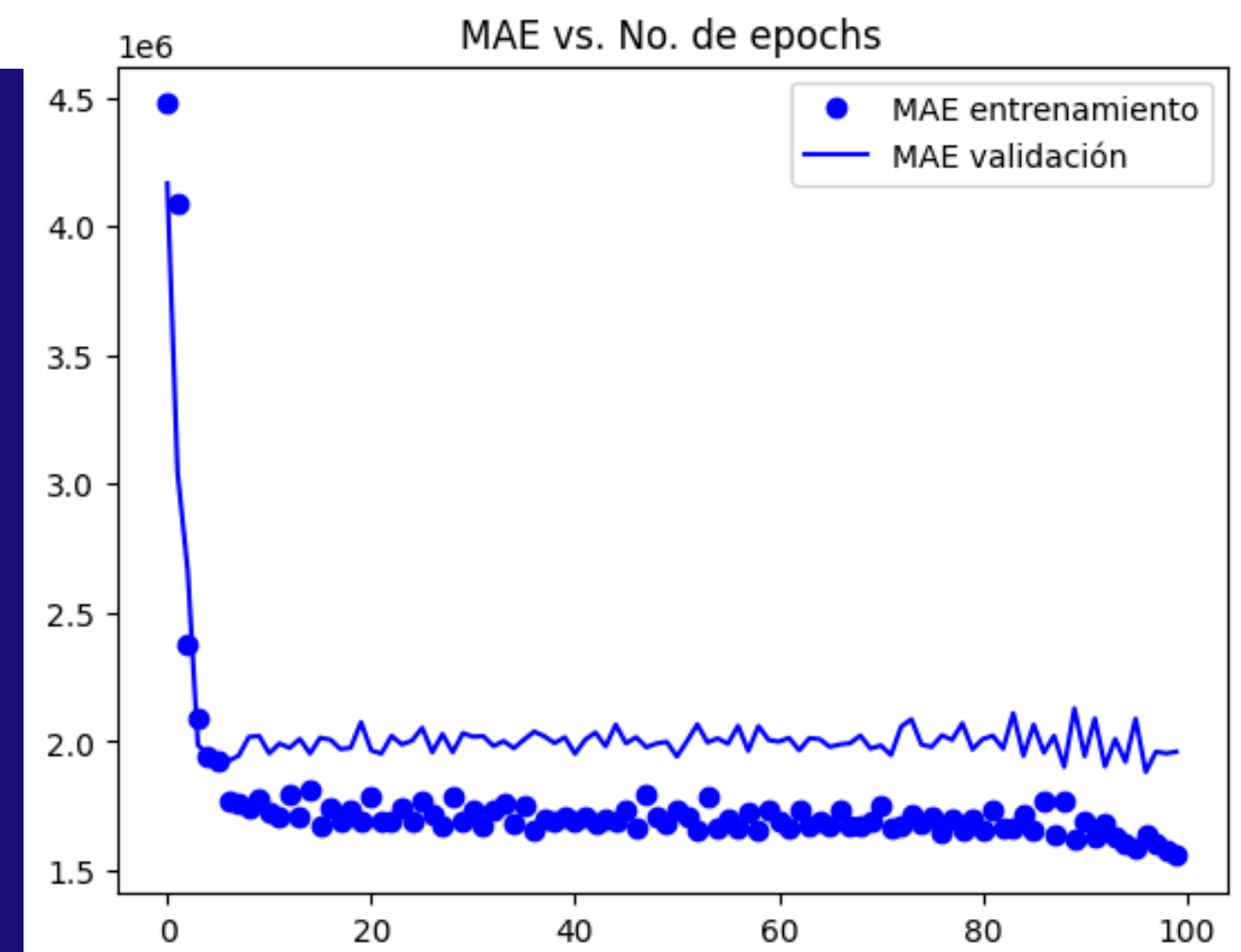
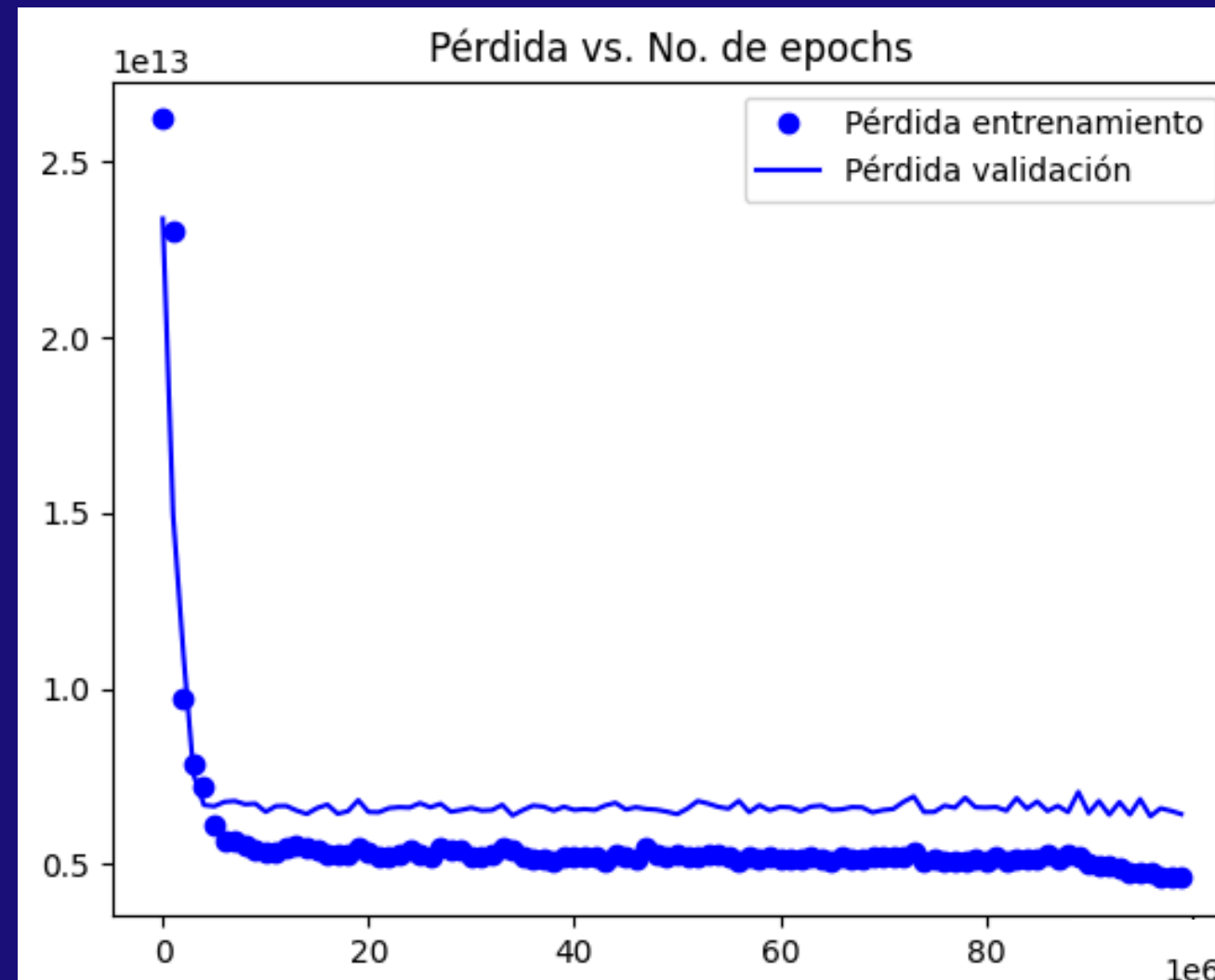


TRAINING

Training Process:

- The model is trained for 100 epochs.
- A batch size of 32 is used.
- Both training and validation sets are included to evaluate performance at each epoch.

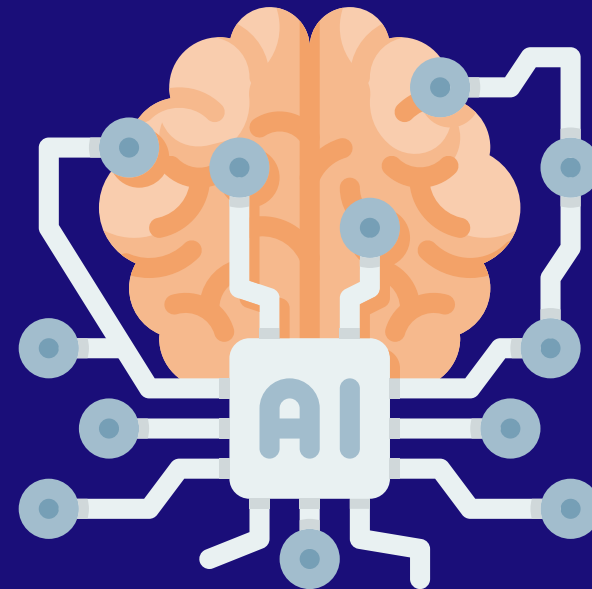
Graphics



MODEL EVALUATION

Metric evaluation:

- Mean Squared Error (MSE): Calculated value
- Mean Absolute Error (MAE): Calculated value
- These values indicate the performance of the model in predicting salary on the test set.



ANALYSIS

1. Loss vs. Number of Epochs:

- **Initial Observation:** There is a rapid decrease in loss in the first few epochs, which is normal as the model starts learning quickly at the beginning.
- **Stabilized Value:** After the first 10 -15 epochs, the loss seems to stabilize on both the training and validation sets. However, the values at which the loss stabilizes are extremely high, indicating that the model is not fitting the data properly.
- **Loss Range:** The loss stabilizes in the range of 0.5×10^{13} , suggesting that the errors in the predictions are large.

2. MAE (Mean Absolute Error) vs. Number of Epochs:

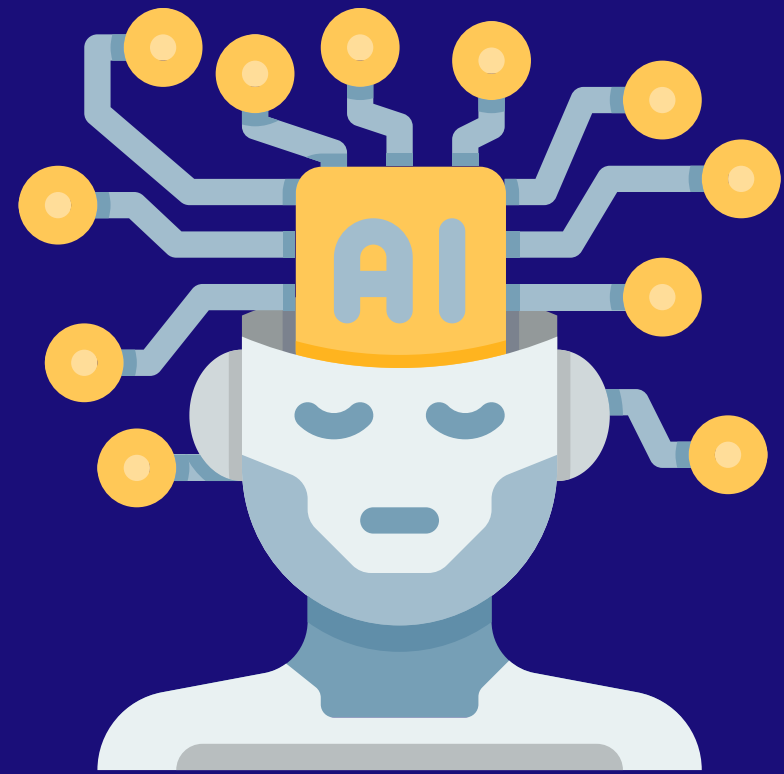
- **Initial Observation:** Similar to the loss graph, the MAE decreases rapidly at first and then stabilizes.
- **Stabilized Value:** The MAE stabilizes around 1.5×10^6 for training and somewhat higher for validation, which is still a high value.

- **Numerical Results:**

MSE (Mean Square Error): 6.42×10^{12}

MAE (Mean Absolute Error): 1.96×10^6

These values suggest that the model predictions have, on average, an error of around 1.96 million monetary units, which is considerably high. And it is used for quadratic data.



Thank You