DLSS Assignment 1

Kuon Ito

May 2025

1 Introduction

The objective of this project is to predict the salary (Sueldo) based on a dataset of job vacancies containing 29 explanatory variables. The prediction task is formulated as a supervised regression problem.

To address this, a structured pipeline was designed. The dataset was first preprocessed to handle missing values and ensure compatibility with machine learning models. The data was then split into training, validation, and test subsets. Three different Multi-Layer Perceptron (MLP) models were built and trained using the training and validation sets, each varying in architecture and hyperparameters.

The performance of each model was primarily evaluated using Mean Absolute Percentage Error (MAPE), as it provides a relative measure of prediction error that is easy to interpret. The model with the lowest MAPE on the validation set was selected for final testing on the held-out test set. Finally, the performance of the best MLP model was compared with a standard Linear Regression baseline to assess the added value of using deep learning techniques.

2 Results

2.1 Data

The dataset consists of 28,631 job vacancy records, each described by 29 input features and one target variable, Sueldo, which represents the salary. The features include a mix of numerical variables (such as edad, experiencia) and categorical variables (such as departamento, tipo_de_trabajo, and nivel_educativo).

Initial exploration revealed that the target variable Sueldo spans a wide range of values, with a long right tail, indicating significant skewness. This was confirmed through histogram plots, motivating the use of a logarithmic transformation on the salary to reduce skew and improve model stability. Furthermore, several features contained missing values. For numerical columns, missing values were imputed with the median to reduce the impact of outliers, while categorical columns were filled using the mode.

To make the data compatible with machine learning algorithms, categorical features were converted using one-hot encoding, which increased the dimensionality but allowed the model to handle non-numeric input.

The dataset was split into three distinct subsets: 70% for training, 15% for validation, and 15% for testing. Random state was set as 42.

All numeric features were standardized using a StandardScaler, ensuring zero mean and unit variance to prevent scale-related dominance among input features.

Finally, the preprocessed features and target were converted into PyTorch tensors.

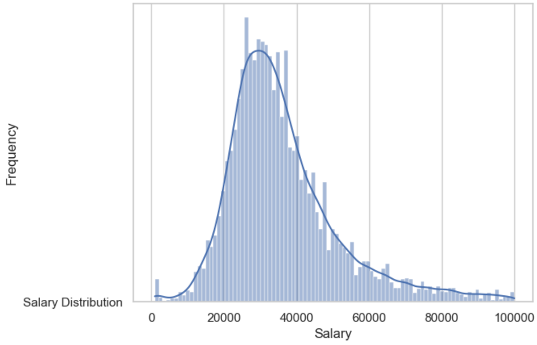

Figure 1: The distribution of Sueldo

2.2 Models

I made three Multi-Layer Perceptron models with varying architectures and hyperparameters. For the activation functions in the hidden layers, ReLU (Rectified Linear Unit) and LeakyReLU were used. ReLU is widely adopted in deep learning due to its simplicity and computational efficiency, and it helps mitigate the vanishing gradient problem. However, since ReLU can lead to inactive neurons (dying ReLU problem), LeakyReLU was also explored in alternative models to allow a small gradient flow even when activations are negative, which can improve robustness and stability during training. For the output layer, no activation function was applied. Since this is a regression task that requires the model to predict continuous values without range restrictions, a linear output (i.e., no activation) is the most appropriate choice.

Regarding the loss functions, both Mean Squared Error (MSE) and Smooth L1 Loss were used. MSE is a standard loss function for regression tasks and is effective when the target distribution is relatively clean. However, because salary data can often include outliers or large deviations, Smooth L1 Loss—also known as Huber Loss—was tested as it is less sensitive to outliers, combining the advantages of both L1 and L2 losses. This choice makes the model more robust in the presence of noisy or skewed data.

Each model was trained using these combinations, and their performance was compared using Mean Absolute Percentage Error (MAPE) on a validation set to determine the most effective configuration.

Model	**Layers**	**Activation**	**Loss Function**	**Learning Rate**	**Batch Size**	**Epochs**
Model 1	64-8-1	ReLU, ReLU	MSELoss	0.0005	128	200
Model 2	128-64-32-1	LeakyReLU, LeakyReLU, LeakyReLU	SmoothL1Loss	0.001	64	100
Model 3	128-64-1	LeakyReLU, LeakyReLU	MSELoss	0.001	64	200

Table 1: Comparison of MLP models

2.3 Training

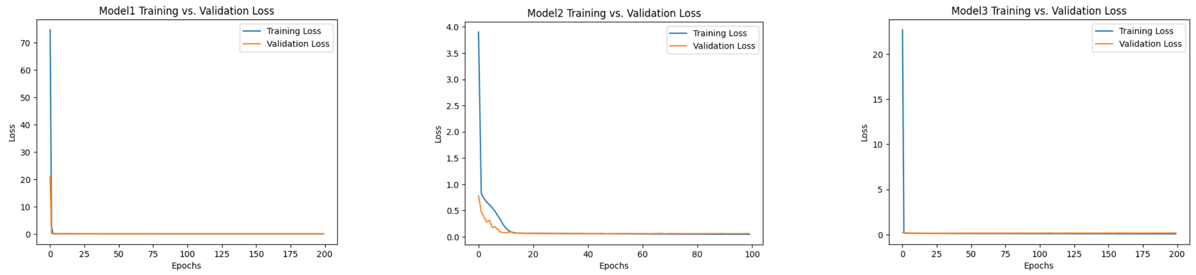

Figure 2: Comparison of the graphs for training vs validation loss

The training and validation loss curves for the three MLP models are shown in Figures 2. Models 1 and 3 demonstrated very rapid convergence, with both training and validation losses stabilizing within the first few epochs. Model 2 required slightly more time to converge, with both losses settling around epoch 20.

In all three models, the training and validation losses converge almost simultaneously and maintain stability thereafter. This consistent alignment between training and validation performance suggests that none of the models suffered from overfitting during training.

In Model 2, the validation loss temporarily dips below the training loss during the early epochs. This is likely due to regularization techniques such as Dropout or Batch Normalization being active during training but not during validation, resulting in more stable predictions on the validation set. Additionally, the difference in how losses are computed—training loss per mini-batch vs. validation loss per epoch—may contribute to this behavior. The temporary nature of this effect suggests it is not indicative of data leakage or overfitting.

Model	**Validation MAPE (%)**
Model 1	2.59
Model 2	2.40
Model 3	2.81

Table 2: Validation MAPE comparison of the three MLP models

The performance of the models was evaluated using the Mean Absolute Percentage Error (MAPE). MAPE was chosen as it provides a scale-independent measure of prediction accuracy, which is particularly useful in regression tasks involving monetary values that span a wide range. Unlike metrics such as Mean Squared Error (MSE), MAPE expresses the error as a percentage, making it easier to interpret and compare across models.

Based on the validation set results (Table2), Model 2 achieved the lowest MAPE score (2.40%), indicating superior predictive performance relative to the other architectures when evaluated using this metric. Therefore, Model 2 was selected as the final model for further testing and comparison.

Model	Test MAPE (%)
Model 1	2.57
Model 2	2.34
Model 3	2.75

Table 3: Test MAPE comparison of the three MLP models

2.4 Evaluation

After evaluating all three models on the validation set using MAPE, Model 2 was selected as the final model due to its lowest validation MAPE (2.40%). It achieved a test MAPE of 2.34%, confirming its strong generalization performance. For reference, the other two models achieved test MAPE scores of 2.57% (Model 1) and 2.75% (Model 3), both higher than Model 2. These results further support the selection of Model 2 as the most effective architecture.

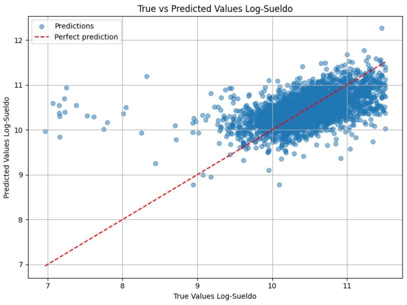

Figure 3: True vs predicted values

Figure3 presents the scatter plot of predicted versus true salary values (in logarithmic scale) on the test set using Model 2. The red dashed line represents the ideal case where the predictions perfectly match the true values. As seen in the figure, most predictions lie close to the diagonal, indicating that the model captures the overall distribution of salaries well.

While some deviation exists, particularly for extreme salary values, the concentration of points around the line suggests strong predictive performance and low systematic bias. This visual evidence aligns with the low MAPE obtained on the test set, further validating the effectiveness of Model 2.

To contextualize the performance of the neural network models, a standard Linear Regression model was trained and evaluated as a baseline. This model achieved a test MAPE of 389.47%, which is significantly worse than the performance of the MLP models. The extremely high error indicates that the linear model was unable to capture the underlying non-linear relationships present in the data.

In contrast, Model 2—our best-performing MLP—achieved a test MAPE of only 2.34%, demonstrating its superior ability to model complex patterns. This substantial gap underscores the effectiveness of using deep learning approaches over traditional linear methods for this salary prediction task.

3 Conclusion and Discussion

Three Multi-Layer Perceptron models were developed and evaluated to predict salaries from a job vacancy dataset. Among the three architectures, Model 2 demonstrated the best overall performance, achieving the lowest validation and test MAPE scores (2.40% and 2.34%, respectively). The consistent performance across validation and test sets indicates strong generalization capabilities.

Model 2 distinguished itself through several key architectural improvements. It incorporated Batch Normalization and Dropout layers, which likely contributed to its stability and resistance to overfitting. Additionally, the use of the LeakyReLU activation function allowed better gradient flow during training, mitigating the dying ReLU problem encountered in simpler models. The loss function used was Smooth L1 Loss, which is less sensitive to outliers than standard MSE, and proved effective given the variability in the target salary values.

Compared to a baseline Linear Regression model, which yielded a test MAPE of 389.47%, the MLP models—and Model 2 in particular—clearly captured the non-linear relationships in the data more effectively.

Overall, this project demonstrates the power of deep learning approaches in structured data regression tasks, especially when appropriate architectural choices and regularization techniques are applied. Further improvements may be achieved by experimenting with alternative architectures, feature engineering, or ensembling methods.

4 Bonus Task 1: Standard Machine Learning

To further evaluate the performance of the proposed MLP models, a Random Forest Regressor was trained as a standard machine learning baseline. The Random Forest achieved a test MAPE of 2.10%, which outperformed all

MLP models, including the best-performing Model 2 (2.34%).

This result suggests that for this particular tabular regression task, tree-based ensemble methods such as Random Forests may be more effective than deep neural networks. Random Forests are known for their robustness and ability to model non-linear relationships without extensive hyperparameter tuning or feature scaling. Nonetheless, the performance of the MLP model remained competitive and demonstrates the potential of neural approaches when properly regularized and tuned.

5 Bonus Task 2: Model Ensemble

An ensemble of three independently trained MLP models was constructed by averaging their output predictions. The ensemble was evaluated on the test set and achieved a MAPE of 2.39%.

The ensemble underperformed compared to the Random Forest Regressor baseline, which achieved a significantly better MAPE of 2.10%. The ensemble also performed slightly worse than the best individual model, Model 2, which achieved a lower test MAPE of 2.34%. This result suggests that, in this case, ensembling did not yield a performance improvement. A possible explanation is that the ensemble included weaker models, which diluted the overall predictive accuracy.

6 Bonus Task 3: Classification

For this task, the continuous salary values were converted into a binary classification target by labeling salaries equal to or greater than 30,000 as class 1 and those below 30,000 as class 0. This threshold resulted in a reasonably balanced dataset, with 62.3% of the samples labeled as class 1 and 37.7% as class 0.

A classification model was implemented using an MLP architecture consisting of four hidden layers with 128, 64, 32, and 16 units respectively. Each hidden layer was followed by a ReLU activation and dropout regularization. The output layer consisted of a single neuron with a Sigmoid activation function to model the probability of the positive class. Early stopping was employed with a patience of 10 epochs to prevent overfitting. The training progression is shown in Figure 4. As shown in Figure 4, both the training and validation loss decrease during the initial stages of training. However, after approximately 10 epochs, the validation loss begins to fluctuate and shows signs of increasing, while the training loss continues to decline steadily.

This divergence suggests the early onset of overfitting. To mitigate this, early stopping with a patience of 10 epochs was employed. This allowed the model to stop training before significant overfitting occurred, ensuring better generalization to unseen data.

The trained MLP classifier demonstrated strong performance on the binary classification task, achieving an accuracy of 76.1%, precision of 78.9%, recall of 85.3%, and an F1 score of 81.9%. When compared to standard machine learning baselines, the MLP outperformed Logistic Regression, which reached an F1 score of 79.1%. However, the Random Forest classifier yielded the highest overall performance, achieving an F1 score of 84.7%, along with superior precision, recall, and accuracy.

These results indicate that while the MLP was effective and competitive, especially compared to linear models, Random Forest can offer even better predictive performance on structured tabular data.

Model	**Accuracy**	**Precision**	**Recall**	**F1 Score**
MLP	0.7614	0.7887	0.8526	0.8194
Logistic Regression	0.7311	0.7656	0.8180	0.7910
Random Forest	0.8019	0.8151	0.8813	0.8469

Table 4: Comparison of classification performance across models

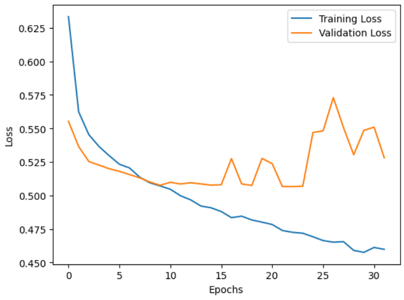

Figure 4: Loss curves for classification