Deep Learning Assignment 1 Model Report

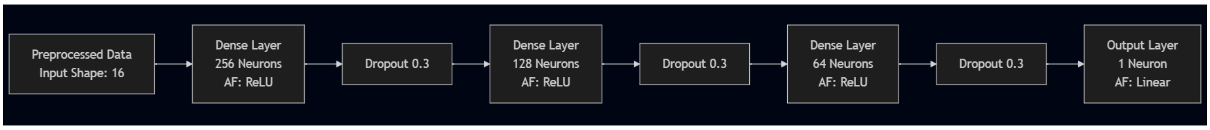
1. Model Architecture and Design Decisions

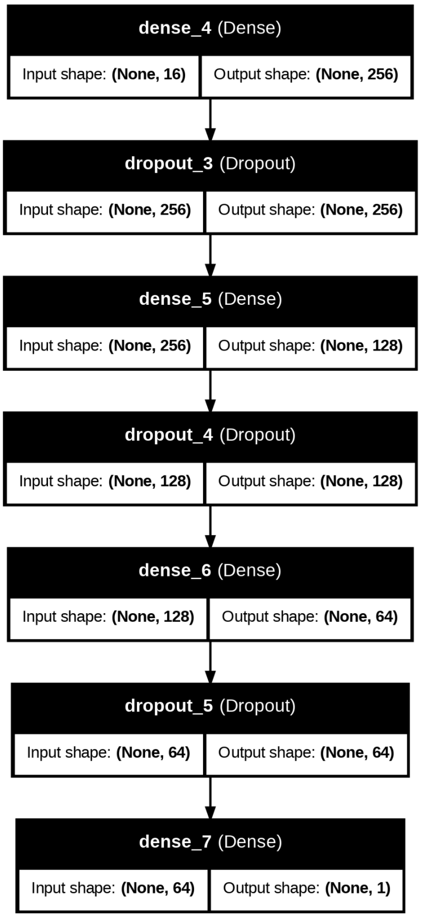
Framework:

- The model is built using **TensorFlow/Keras**, which provides high-level APIs for deep learning.

Model Architecture:

- The architecture consists of multiple layers designed for a regression task to predict sales prices.
- **Input Layer:** The model accepts **16 features** as input, which are numerical and categorical features processed appropriately.
- **Hidden Layers:**
    - **First Hidden Layer:** Contains **256 neurons** with **ReLU (Rectified Linear Unit) activation function**, which helps in learning complex patterns while preventing vanishing gradients.
    - **Dropout (0.3):** A regularization technique is applied to randomly set 30% of the neurons to zero during training to prevent overfitting.
    - **Second Hidden Layer:** Contains **128 neurons** with **ReLU activation** and **Dropout (0.3)**.
    - **Third Hidden Layer:** Contains **64 neurons** with **ReLU activation** and **Dropout (0.3)**.
- **Output Layer:**
    - The final layer has **1 neuron** with **linear activation**, which is suitable for regression problems as it directly outputs a continuous value (predicted sales price).

- **Architecture** :
- 

  - 

Optimization and Loss Function:

- **Optimizer:** Adam (Adaptive Moment Estimation), which adapts the learning rate dynamically based on past gradients, leading to faster and more stable convergence.
- **Learning Rate:** 0.001, which balances training speed and stability.
- **Loss Function:** Mean Squared Error (MSE), as it penalizes large errors more than smaller ones, making it suitable for regression.
- **Metrics:** Mean Absolute Error (MAE), which provides an interpretable error metric in the same unit as the target variable.

2. Training Process

Preprocessing Steps:

- **Feature Scaling:**
  - Since deep learning models perform better with standardized data, **StandardScaler** is applied to scale both features and the target variable.
  - Scaling ensures that all numerical values have a mean of 0 and a standard deviation of 1, improving gradient descent efficiency.

Training Details:

- **Dataset Split:**
  - The dataset is split into **80% training data** and **20% validation data**.
- **Epochs:**
  - The model is trained for **350 epochs**, meaning it goes through the entire dataset 350 times to learn patterns.
- **Batch Size:**
  - A batch size of **32** is used, meaning updates to the model weights are made after every 32 samples.
- **Early Stopping:**
  - Enabled with **patience = 20**, meaning training stops if validation loss does not improve for 20 consecutive epochs.
  - Restores the best model weights to avoid overfitting.

3. Evaluation Metrics

Evaluation Steps:

- After training, the model is evaluated on the test dataset using MSE and MAE.
- The predictions are transformed back to their original scale (since they were scaled earlier).

- **R2 Score** is computed to measure how well the model explains the variance in the target variable.

Key Metrics:

- **Mean Absolute Error (MAE):** Represents the average absolute difference between predicted and actual values. A lower MAE indicates better performance.

- **R2 Score:** Measures how well the model explains the variance in sales price predictions. A value closer to 1 indicates a better model.

4. Training Process and Loss Curves

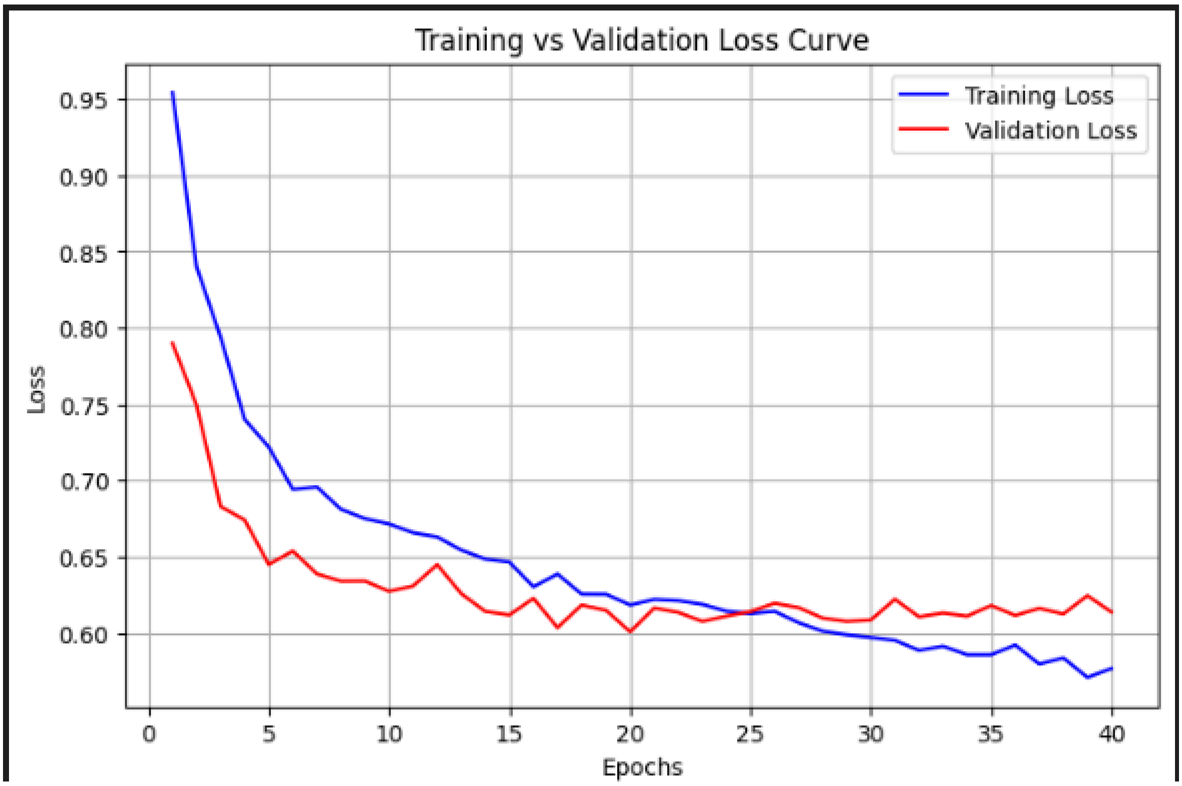
- The training process involved monitoring both training and validation loss.

- A plot of the loss curve shows that:

  - The training loss decreases steadily, indicating that the model is learning.

  - The validation loss initially decreases but plateaus, showing early stopping prevented overfitting.

- The model generalizes well to unseen data, as indicated by stable validation metrics.

- 

5. Conclusion

- The deep learning model successfully learned to predict sales prices with reasonable accuracy.
- Further improvements could be achieved by:
    - Experimenting with different architectures (e.g., additional layers or neurons).
    - Fine-tuning hyperparameters such as dropout rates, batch sizes, and learning rates.
    - Exploring alternative feature engineering techniques to improve input data representation.

This report provides an in-depth analysis of the model’s architecture, training process, evaluation, and future enhancements to improve performance.