**Artificial Neural Network (ANN) Architecture for Predicting Customer Churn**

**1. Objective** The goal of this Artificial Neural Network (ANN) model is to accurately predict the likelihood of customer churn in a telecommunications company, using structured customer data with both numerical and categorical features.

**2. Input Features**

* The model was trained on a preprocessed and encoded dataset derived from customer data.
* After one-hot encoding and feature engineering, the final dataset contained 20 input features (e.g., tenure, MonthlyCharges, TotalServiceCost, Contract type, Payment method, etc.).

**3. ANN Architecture**

The Artificial Neural Network was designed with multiple layers to enable it to learn complex, non-linear patterns in the data. Each layer has a specific role in transforming the input data into a meaningful prediction:

**3.1 Input Layer**

* **Units:** 20 neurons (each representing one input feature)
* **Activation:** None. This layer serves only to accept input values which are then passed to the next layer without transformation.

**3.2 Hidden Layer 1**

* **Units:** 64 neurons
* **Activation:** ReLU (Rectified Linear Unit), which introduces non-linearity and helps the model learn from complex relationships between variables.
* **Dropout:** A 30% dropout rate randomly disables 30% of neurons during training, reducing the chance of overfitting by forcing the network to learn redundant representations.

**3.3 Hidden Layer 2**

* **Units:** 32 neurons
* **Activation:** ReLU
* **Dropout:** Another 30% dropout to maintain regularization and robustness.

**3.4 Output Layer**

* **Units:** 1 neuron
* **Activation:** Sigmoid. This transforms the output to a probability value between 0 and 1, which can be interpreted as the likelihood of churn.

**4. Optimization and Training Details**

To ensure effective training, we used several optimization strategies:

* **Loss Function:** Binary Crossentropy, which is ideal for binary classification tasks like churn prediction. It quantifies the difference between the predicted probabilities and the actual class labels.
* **Optimizer:** Adam (Adaptive Moment Estimation) optimizer was used for its ability to adjust learning rates automatically and perform well with sparse gradients.
* **Metrics:** Multiple metrics including Accuracy, Precision, Recall, and F1-Score were tracked to evaluate both general and class-specific performance.
* **Epochs:** The model was trained over a maximum of 100 epochs. However, it typically stopped earlier due to early stopping.
* **Batch Size:** 32 records per training iteration, which strikes a balance between model stability and training speed.
* **Early Stopping:** A strategy to monitor validation loss and halt training if no improvement is seen after 5 consecutive epochs, preventing overfitting and saving computational time.

**5. Evaluation & Performance**

The model was evaluated using a separate test dataset to assess its generalization performance:

* **Accuracy:** Achieved 73%, indicating that the model correctly predicted churn status in nearly three out of four cases.
* **Recall (Churn Class):** Improved to 0.79, meaning the model correctly identified 79% of customers who actually churned. This is critical for retention strategies.
* **Precision:** Though slightly lower, it reflects the proportion of predicted churns that were correct—indicating some trade-off for higher recall.
* **F1-Score:** Balanced between precision and recall, providing a single metric that considers both false positives and false negatives.

These metrics suggest the model is especially good at detecting churn cases, which is valuable for proactive intervention.

**6. Summary** This ANN model architecture was selected and fine-tuned to balance performance and generalization. Techniques like dropout, early stopping, and class weighting were used to manage overfitting and class imbalance. This predictive model can now support the business in targeting retention strategies more effectively.