**Stage 3 – Predictive Modeling and Customer Churn Analysis**

**Overview**

In Stage 3, we focused on building a robust predictive model to identify customers likely to churn. After completing data preprocessing and exploratory analysis in earlier stages, we transitioned to predictive modeling using an Artificial Neural Network (ANN). This stage involved defining the ANN architecture, training it on scaled and encoded data, evaluating its performance, and interpreting key findings.

**1. Artificial Neural Network (ANN) Architecture**

We used a feedforward neural network, tailored to capture the complex relationships within our customer data. The ANN model comprises the following layers:

**Input Layer**

* Contains 20 neurons, each representing a distinct input feature (e.g., tenure, MonthlyCharges, Contract type, etc.).
* No activation function is applied here; it serves to pass the scaled inputs to the next layer.

**Hidden Layer 1**

* Consists of 64 neurons.
* Uses the ReLU (Rectified Linear Unit) activation function, which introduces non-linearity and avoids the vanishing gradient problem common in deep networks.
* Includes a Dropout layer (30%) to randomly deactivate neurons during training. This regularization technique helps prevent overfitting by forcing the network to learn multiple independent representations.

**Hidden Layer 2**

* Contains 32 neurons with ReLU activation.
* Another Dropout layer (30%) is used to continue regularization as model depth increases.

**Output Layer**

* Contains 1 neuron.
* Uses a Sigmoid activation to produce a probability between 0 and 1, indicating the likelihood of customer churn.

The chosen architecture balances depth and complexity to avoid overfitting while enabling accurate classification of churn behavior.

**2. Optimization and Training Details**

To ensure the model learns efficiently and generalizes well, we applied several strategies during training:

**Loss Function**

* We used Binary Crossentropy, ideal for binary classification tasks. This function penalizes incorrect predictions more as they deviate from the true class probability.

**Optimizer**

* The Adam (Adaptive Moment Estimation) optimizer was used. It combines the strengths of both RMSProp and SGD with momentum, adjusting learning rates dynamically based on the gradient’s first and second moments.

**Training Configuration**

* Batch Size: 32 – offering a balance between noise (which helps generalization) and computational efficiency.
* Epochs: Capped at 100, but we used EarlyStopping to monitor validation loss. Training was halted if no improvement was observed after 5 consecutive epochs, reducing the risk of overfitting.
* Validation Split: A small portion (typically 20%) of the training set was reserved for validation during training to assess performance and trigger early stopping.

**Class Imbalance Handling**

* Churned customers were underrepresented in the dataset. We addressed this imbalance by assigning class weights, giving more importance to churn cases to reduce bias in learning.

**3. Evaluation and Performance Analysis**

Once the model was trained, we assessed its performance using the test dataset. Evaluation was done using both confusion matrix metrics and classification reports.

**Confusion Matrix Output:** [[707 284] [ 76 282]]

* True Negatives (707): Non-churners correctly classified
* False Positives (284): Non-churners incorrectly classified as churners
* False Negatives (76): Churners missed by the model
* True Positives (282): Churners correctly identified

**Performance Metrics:**

* Accuracy: 73%  
  Indicates the overall percentage of correctly predicted cases.
* Precision (Churn): 0.50  
  Reflects how many predicted churns were correct. Lower precision suggests the model errs on the side of over-predicting churn, which may be acceptable in a retention-focused context.
* Recall (Churn): 0.79  
  Indicates the model caught 79% of all actual churners. This high recall is critical because missing a churner is more costly than mistakenly targeting a non-churner.
* F1-Score: 0.61  
  Balances both precision and recall into one metric. Given the class imbalance, this is a fair score, especially considering the recall strength.

**Why Recall Matters Here** In churn analysis, recall for the churn class is more important than overall accuracy because:

* Failing to identify a customer who will churn means missed opportunity to intervene.
* The business impact of one customer leaving is higher than wrongly targeting a loyal one.

**4. Insights & Recommendations**

**Churn Drivers Identified:**

* Customers with monthly contracts
* Short tenure users
* Those using electronic check payment methods
* Higher total service cost

**Recommendations:**

* Loyalty Discounts: Offer retention incentives to short-tenure customers.
* Contract Upgrades: Encourage migration to annual or bi-annual plans.
* Payment Options: Simplify and incentivize the use of automatic payments or bank transfers.
* Service Feedback: Reach out to customers with usage patterns associated with churn.

**5. Limitations & Future Improvements**

* The model behaves as a black box, with limited interpretability.
* Performance is subject to the quality and richness of the input dataset.
* Further improvements could include:
  + Feature importance via SHAP or LIME
  + Experimenting with ensemble models
  + Incorporating time-series behavior like usage over months