Table Of Content

1. Objective

Building predictive models to identify customers who are likely to leave a telecom service is the goal of the project. Accurate churn prediction enables telecom businesses to identify at-risk customers early and proactively improve retention rates through targeted interventions.

2. Dataset Overview

- Source: Pre-processed dataset (Resampled_Training_Data.csv) after handling class imbalance.
- Target Variable: Churn_1
    - 1: Customer has churned
    - 0: Customer has not churned
- Key Features:
    - tenure: Duration with the company
    - MonthlyCharges: Monthly billing amount
    - Contract, InternetService, PhoneService: Service type indicators
    - Other encoded categorical and numerical features

3. Data Preprocessing

- Target & Feature Split:
    - X (features): All columns except Churn_1
    - y (target): Churn_1
- Scaling:
    - StandardScaler was used to normalize features for better convergence of the ANN.
- Train-Test Split:
    - 80% Training, 20% Testing (random_state=42)
    - Maintains consistent data distribution.

- Class Imbalance Handling:
    - Applied class_weight='balanced' using sklearn.utils.class_weight to assign higher penalty to the minority class during training.

4. Model Development

The project uses a Sequential Artificial Neural Network (ANN) with the following advanced practices:

Layer Architecture

- Input Layer: Auto configured using Input() based on feature shape.
- Dense Hidden Layers (1–3 layers):
    - Tuned units from 32 to 256.
    - Used LeakyReLU activation to avoid dead neurons.
    - BatchNormalization to stabilize training.
    - Dropout (0.2–0.5) to prevent overfitting.
    - L2 regularization to constrain weights.
- Output Layer:
    - 1 Neuron with Sigmoid activation (binary classification)

5. Hyperparameter Tuning

Used Keras Tuner with RandomSearch to optimize key hyperparameters:

Parameter	Values Explored
Hidden units	64–512
Dropout	0.2–0.5
Number of Layers	1–3
Learning Rate	0.01, 0.001, 0.0005
L2 Regularization	0.001, 0.01, 0.1

Tuning Strategy:

- max_trials = 10 (combinations)
- executions_per_trial = 1

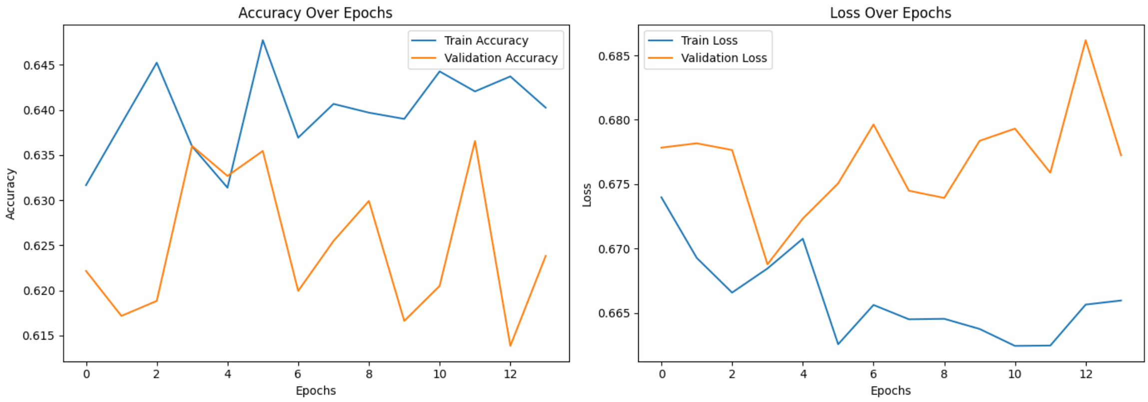
- EarlyStopping(patience=10) to prevent overfitting

6. Model Training

The best model from tuning was trained further for up to 50 epochs. Early stopping was used to halt training if no validation improvement occurred.

Training Performance:

- Accuracy & Loss Curves:
  Training and validation metrics over epochs showed convergence and no significant overfitting.



7. Evaluation Metrics

The final model was evaluated on the test set using the following metrics:

Metric	Description
Accuracy	Overall correctness of predictions
Precision	% of predicted churns that were correct
Recall (Sensitivity)	% of actual churns identified
F1-Score	Harmonic mean of precision and recall
AUC Score	Probability the model ranks a random positive higher than a negative

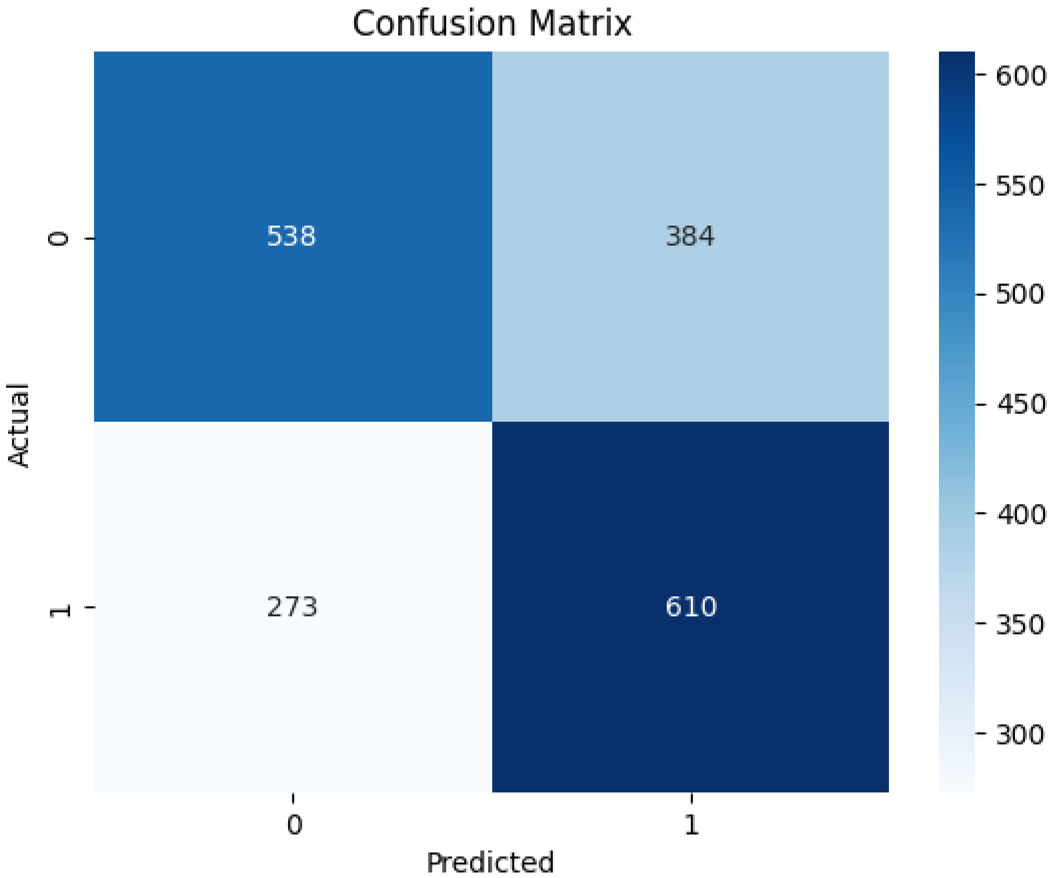
Classification Report:

- Shows detailed precision, recall, and F1-score for both classes (0 and 1).

- Strong balance between recall and precision confirms usefulness in churn targeting.

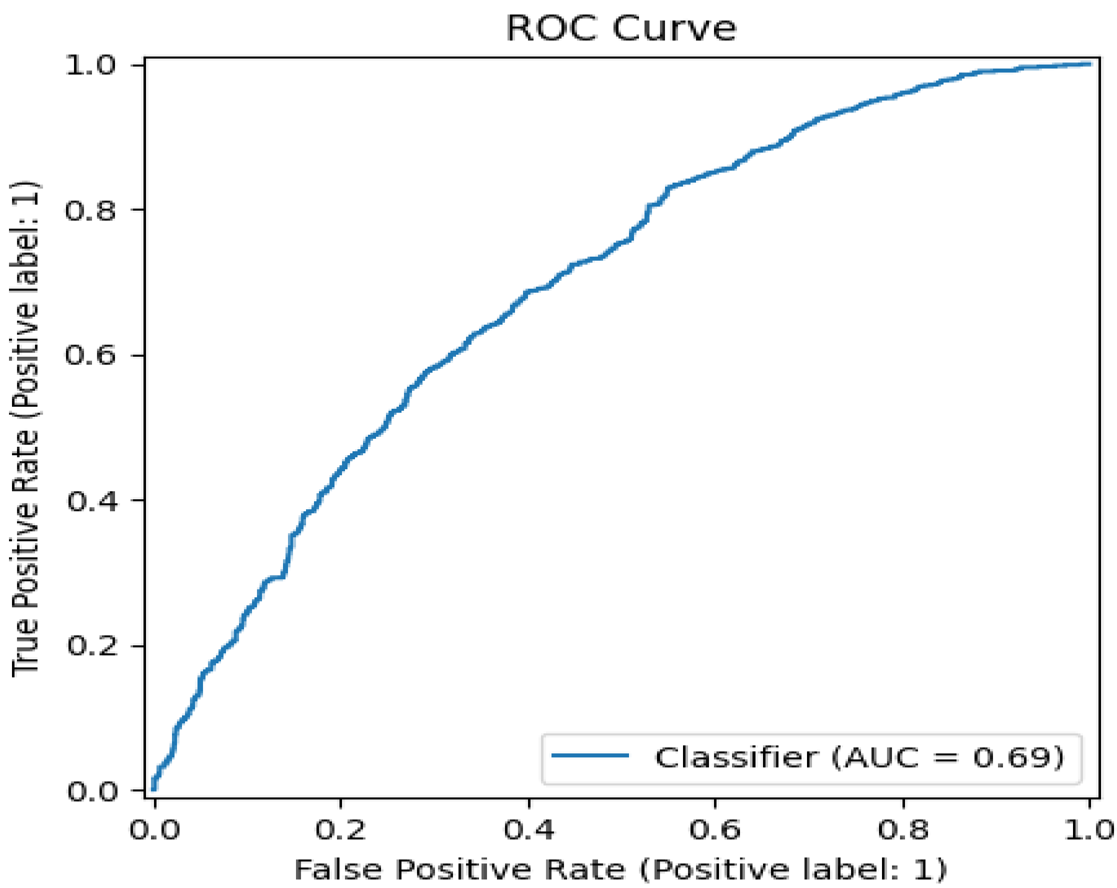
Confusion Matrix:

- Highlights True Positives, True Negatives, False Positives, and False Negatives.



ROC Curve:

- The Area Under Curve (AUC) > 0.80 signifies high-quality separability between churn and non-churn classes.


ROC Curve
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate (Positive label: 1)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1)
Classifier (AUC = 0.69)