```
============================================================
INFERENCE RESULTS
============================================================

Accuracy: 0.9009
AUC Score: 0.9275

Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.94      0.94      2589
           1       0.74      0.75      0.75       619

    accuracy                           0.90      3208
   macro avg       0.84      0.85      0.84      3208
weighted avg       0.90      0.90      0.90      3208


Confusion Matrix:
[[2423  166]
 [ 152  467]]

Confusion matrix plot saved to: data/output/confusion_matrix.jpg
Normalized confusion matrix plot saved to: data/output/confusion_matrix_normalized.jpg
Predictions saved to: data/output/predictions.csv
```

Executive Summary

In this submitted model, I predicted ProdTaken (0/1) using 3208 rows and 19 columns. I cleaned the data by removing duplicates, fixing text errors (example fe male → female, unmarried → single), converting unknown → missing, and filling missing values (numeric = median, categorical = mode). I split the data into 80% train / 20% test, and used validation split 20% with early stopping. The final inference result is Accuracy = 0.9009, AUC = 0.9275, and class 1 recall is 0.75 with F1 = 0.75.

1. Data Preparation

I used the tourism customer dataset **class5data_clean_final.csv** and the target column is **ProdTaken** (0 = not taken, 1 = taken). I cleaned the dataset by removing duplicate rows, fixing messy spacing in text, converting “unknown” values into missing values, and then filling missing values (numeric columns filled by median, text columns filled by the most common value). I also fixed category typos to make the text consistent, such as **“fe male”** → **“female”** and **“unmarried”** → **“single”**, so the model will not treat the same meaning as different categories. After cleaning, I split the dataset into **train and test (80/20)** using stratification so the class ratio stays similar in both sets, and during training I used a **validation split (20% of the training set)** for early stopping.

2. Analysis (what I observe in the data)

The dataset is **imbalanced**. From the label counts, class 0 has **2589** rows and class 1 has **619** rows, so the model will naturally be better at predicting class 0. This explains why the model performance for class 0 is higher than class 1. Because class 1 is smaller, it is easier for the model to miss some true positives (false negatives), which lowers recall and F1-score for class 1.

3. Feature extraction

This dataset contains both numeric and categorical columns. For categorical columns (example: Gender, Occupation, ProductPitched, MaritalStatus), I used **one-hot encoding** so the neural network can use them as numeric inputs. For numeric columns (example: Age, MonthlyIncome), I used **StandardScaler** to normalize the values, because features have different scales and standardization helps training become more stable and faster. This feature extraction step lets the model use both numeric and category information together.

4. Building model (model structure, optimizer, loss, tuning)

I built a binary classification model using **Keras Sequential**. The model uses several **Dense layers** with **ReLU** activation to learn patterns from the features, and I used **Dropout** to reduce overfitting. The output layer uses **Sigmoid** because the output is a probability of ProdTaken = 1. I trained with the **Adam optimizer** and **Binary Crossentropy loss**, which is standard for binary classification. I also used **EarlyStopping** based on validation loss and restored the best weights to prevent over-training.

5. Evaluation results

On the test data, the inference results are:

- **Accuracy = 0.9009**
- **AUC = 0.9275**

From the classification report:

- Class 0: precision **0.94**, recall **0.94**, F1 **0.94**
- Class 1: precision **0.74**, recall **0.75**, F1 **0.75**

The confusion matrix is:

$$\begin{bmatrix} 2423 & 166 \\ 152 & 467 \end{bmatrix}$$

This means the model predicted **2423 true negatives** and **467 true positives**, with **166 false positives** and **152 false negatives**. Overall accuracy is strong, but class 1 performance is lower mainly because the dataset is imbalanced, so improving class 1 recall would be the main next improvement goal.