Machine Learning Pipeline

1. Data Preparation

Firstly, I cleaned the dataset by:

- Checking for misspelled categorical values
- Handling blank or missing cells
- Ensuring consistent data formatting

After cleaning, I split the dataset into **75% training and 25% testing using code**.
I used a stratified **split** to maintain the same proportion of class 0 and class 1 in both datasets.
This is important because the dataset is slightly imbalanced.

The datasets were saved as:

- `train.csv`
- `test.csv`

This step ensures fair evaluation and prevents data leakage.

---

2. Problem Analysis

Before building the model, I analyzed the business objective.

I identified that:

- The target variable is **ProdTaken**
- Class 0 = Customer did not purchase
- Class 1 = Customer purchased

Therefore, the model goal is to correctly predict whether a customer will take the product.

This step is important because understanding the objective helps determine:

- Which features to focus on
- Which evaluation metric matters most

---

3. Feature Extraction

To improve model performance, I performed feature selection and feature engineering.

I:

- Identified impactful features based on domain understanding
- Created additional derived features where necessary
- Removed features that were not relevant or redundant

Feature engineering was done to:

- Increase predictive power
- Improve model learning capability
- Reduce noise

As a result, the dataset became more informative for training.

```
# ── Feature engineering ──────────────────────────────────────────
X['NumberOfChildrenVisiting'] = X['NumberOfChildrenVisiting'].fillna(0)
X['IncomePerTrip']        = X['MonthlyIncome']          / (X['NumberOfTrips'].replace(0, 0.1))
X['PitchPerFollowup']     = X['DurationOfPitch']        / (X['NumberOfFollowups'].replace(0, 0.1))
X['TotalVisitors']        = X['NumberOfPersonVisiting'] + X['NumberOfChildrenVisiting']
X['IncomePerPerson']      = X['MonthlyIncome']          / (X['NumberOfPersonVisiting'].replace(0, 0.1))
X['SatisfactionXIncome']  = X['PitchSatisfactionScore'] * X['MonthlyIncome']
# Extended interactions
X['PassportXIncome']      = X['Passport']               * X['MonthlyIncome']
X['TripsXVisitors']       = X['NumberOfTrips']          * X['NumberOfPersonVisiting']
X['PitchXSatisfaction']   = X['DurationOfPitch']        * X['PitchSatisfactionScore']
X['FollowupXSatisfaction']= X['NumberOfFollowups']      * X['PitchSatisfactionScore']
X['ChildrenRatio']        = X['NumberOfChildrenVisiting'] / (X['TotalVisitors'].replace(0, 0.1))
X['IncomePerTrip2']       = X['MonthlyIncome']          / (X['NumberOfTrips'].replace(0, 0.1) ** 2)
print(f"Features after engineering: {X.shape[1]}")
```

---

4. Model Building

For model development:

- I used class4training materials as a structural reference
- Adjusted preprocessing and modeling steps according to the dataset

- Tuned the pipeline to match the prediction objective

The model was trained using the prepared training dataset and optimized.

---

5. Evaluation

To ensure proper evaluation, I measured:

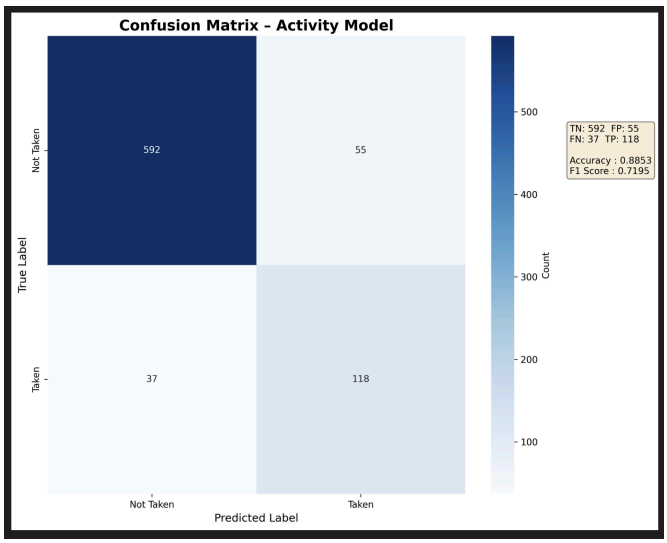
- Accuracy
- Precision
- Recall
- F1-score

Training Performance

- Accuracy: **0.9170**
- Precision: **0.7624**
- Recall: **0.8280**
- F1 Score: **0.7938**

Test (Inference) Performance

- Accuracy: **0.8853**
- Precision: **0.6821**
- Recall: **0.7613**
- F1 Score: **0.7195**



Performance Analysis

Firstly, the training accuracy is 91.70%, while test accuracy is 88.53%. The gap is small, which indicates that the model generalizes reasonably well.

Secondly, recall remains relatively high in both training (0.8280) and testing (0.7613). This means the model successfully identifies a large portion of customers who actually take the product.

However, precision decreased from 0.7624 (train) to 0.6821 (test). This indicates that some false positives occur in unseen data.

The F1-score of 0.7195 on test data shows a balanced trade-off between precision and recall.

Overall:

- There is **no severe overfitting**
- The model maintains **stable performance**
- The generalization gap is acceptable