Customer Booking Prediction for British Airways

Using Machine Learning to Predict Booking Completion

Project Overview

- **Objective:** Predict whether a customer will complete a booking on British Airways based on behavioural and contextual data.
- **Approach:** Built a machine learning model using a customer booking dataset and evaluated performance with accuracy and SMOTE techniques.
- **Business Impact:** Enables targeted marketing, optimizes sales funnels, and enhances customer experience.

Dataset Overview

- **Source:** Simulated booking data (50,000 rows)
- **Goal:** Predict booking complete (0 or 1)
- **Target Variable:** booking complete
- **Features Include:**
  - Customer preferences: wants extra baggage, wants preferred seat, wants in flight meals
  - Booking behavior: purchase lead, length of stay, flight hour, flight day
  - Context: sales channel, trip type, booking origin, route

Data Cleaning & Preprocessing

- No null values — ✅
- Converted categorical features like “flight day” manually (Mon=1, ..., Sun=7)
- **Boolean Features Converted:**

Columns “**wants extra baggage**”, “**wants preferred seat**”, “**wants in flight meals**” converted to integers (0 or 1) for compatibility with ML models

Data Cleaning & Preprocessing

Column Transformer Setup:

- Used StandardScaler on numerical features
- Applied OneHotEncoder on categorical features with handle unknown='ignore’
- Combined using **ColumnTransformer** to create a unified preprocessing pipeline

Exploratory Data Analysis (EDA)

Key Statistical Summary of Numerical Features:

(Next Slide)

Feature	Mean	Std Dev	Min	25%	Median	75%	Max
num passengers	1.59	1.02	1	1	1	2	9
purchase lead	84.94	90.45	0	21	51	115	867
length of stay	23.04	33.89	0	5	17	28	778
flight hour	9.07	5.41	0	5	9	13	23
flight day	3.81	1.99	1	2	4	5	7
wants extra baggage	0.67	0.47	0	0	1	1	1
wants preferred seat	0.30	0.46	0	0	0	1	1
wants in flight meals	0.43	0.49	0	0	0	1	1
flight duration	7.28	1.50	4.67	5.62	7.57	8.83	9.5
booking complete	0.15	0.36	0	0	0	0	1

Insights from table

- **Highly Imbalanced Target:** Only ~15% of records are booking complete = 1. Strong case for using **SMOTE**.
- **purchase lead** is very spread out — some book almost a year in advance, others same day. May signal traveler commitment.
- **length of stay** has a very long tail (up to 778 days!). Consider log transformation or outlier treatment.
- Most travelers are single passengers; group travel is rare.
- wants extra baggage is most popular among preferences; preferred seat is least

Modeling Approach

Created a full Scikit-learn Pipeline with:

- preprocessor: combines:
- StandardScaler for numerical features
- OneHotEncoder for categorical features (with handle unknown='ignore')
- classifier: RandomForestClassifier (n estimators=100, random state=42)

Why Random Forest?

- Robust to overfitting
- Handles both numerical and categorical features
- Automatically ranks feature importance
- Doesn’t require heavy scaling (but scaling helps when used in pipelines with other models)

Notes: End-to-end pipeline improves maintainability and prevents data leakage

Model Evaluation (Before SMOTE)

Model Used: Random Forest (100 trees)

- **Test Set Accuracy: 84.4%**

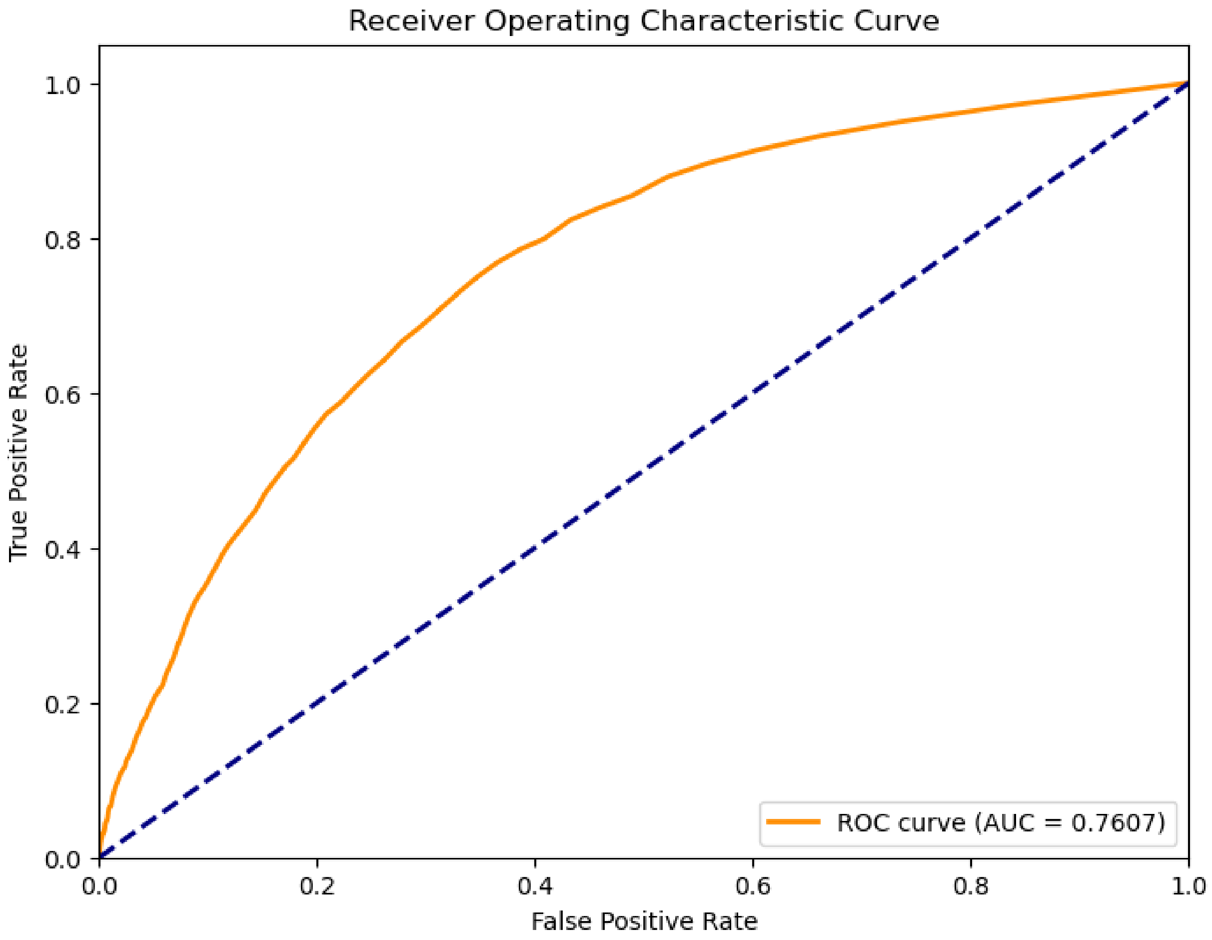
Class	Precision	Recall	F1-score	Support
0 (Not Booked)	0.87	0.96	0.91	12,784
1 (Booked)	0.43	0.17	0.25	2,216

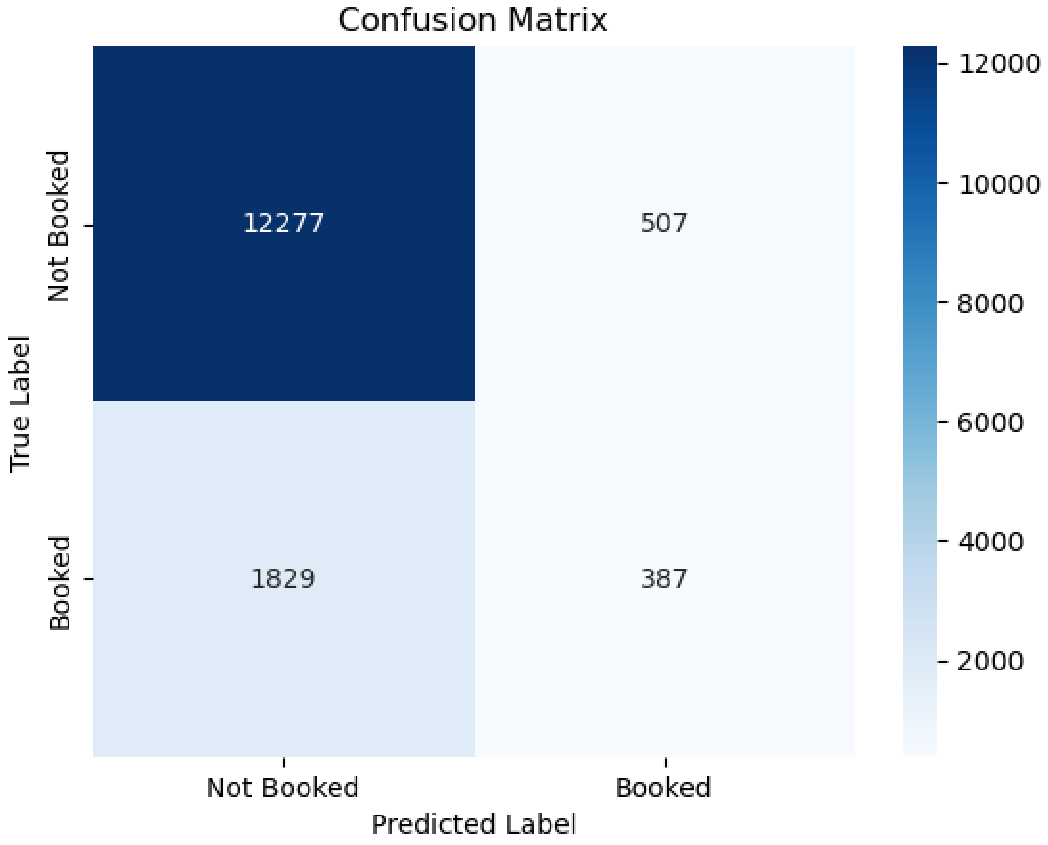
- **Macro Avg F1-score:** 0.58

Imbalance Issue Noted:

- Model favors class 0 heavily
- Very **low recall (0.17)** for actual bookings – many are being missed
- This justifies the **need for SMOTE** (class balancing)

Confusion Matrix Visualization & ROC Curve & AUC Score:





Class Imbalance Handling – Class Weights

Updated Model:

- Random Forest with class weight='balanced’

Why Use Class Weights?

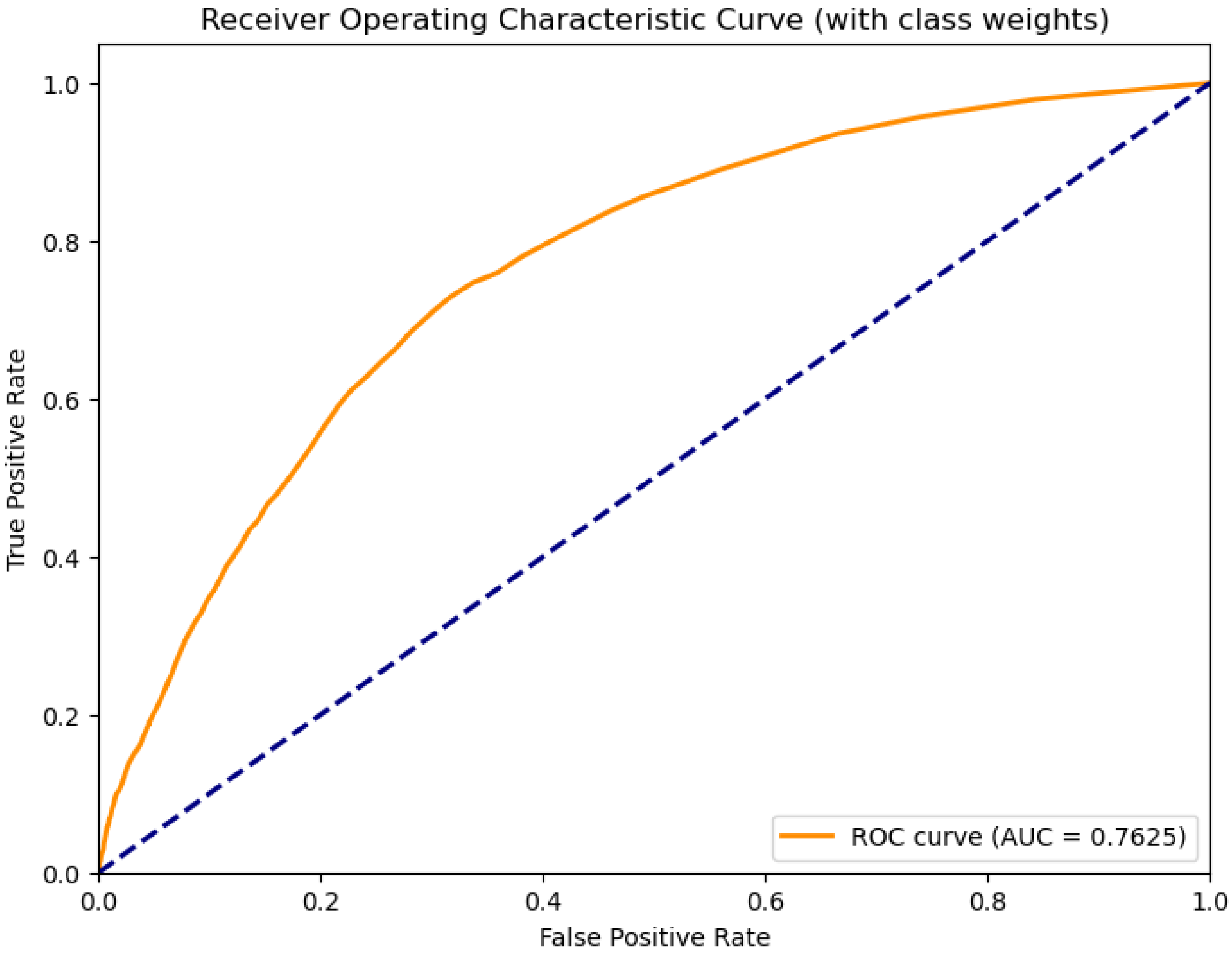
- Automatically assigns higher weight to the minority class (bookings = 1)
- Helps model pay more attention to underrepresented outcomes
- **Results (Class-Weighted Random Forest):**

Class	Precision	Recall	F1-score	Support
0	0.87	0.96	0.91	12,784
1	0.43	0.16	0.24	2,216

•**Accuracy:** 84% (same as before)
•**Minor improvement in class 1 precision**
•**Still very low recall (0.16)** → model still misses many real bookings

ROC Curve:

- **ROC AUC Score: 0.7625**
- Indicates **fair discrimination ability** between bookings and non-bookings
- Better than random (0.5), but still room for improvement
- Confirms that model has potential, but **struggles with true positives (recall)**



SMOTE – Oversampling for Class Balance

What is SMOTE?

- **SMOTE (Synthetic Minority Oversampling Technique)** generates synthetic examples of the minority class to balance the dataset

Why Use It?

- Addresses poor recall from previous models
- Allows the model to see more examples of the rare “booking complete” class

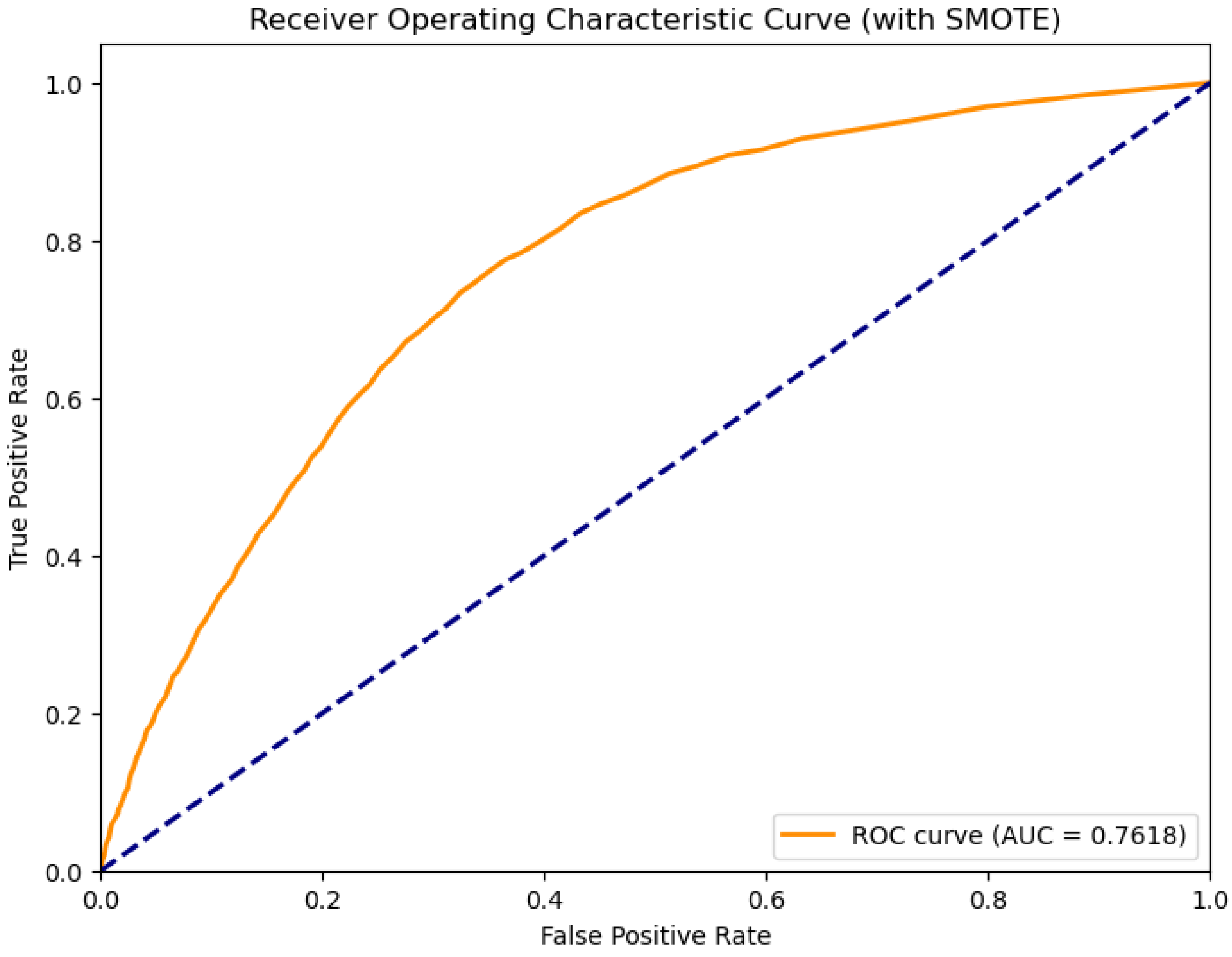
Model Performance – With SMOTE

Class	Precision	Recall	F1-score	Support
0	0.88	0.92	0.90	12,784
1	0.38	0.29	0.33	2,216

- **Accuracy:** 82%
- **ROC AUC Score: 0.76**
- 📈 **Improved recall & F1** for class 1 (bookings) compared to previous models
- Slight accuracy drop, but **better balance** and fairness across classes

ROC Curve (with SMOTE)

- Clean ROC curve plotted with AUC = **0.7618**
- Demonstrates solid class separation ability
- Better than random, and shows improved model performance over previous attempts



Final Thoughts!!!

- Booking completion is highly imbalanced → standard models underperform
- SMOTE improves minority class recall and F1-score
- ROC AUC = 0.76 — good baseline model for production/testing

Thank You!

Project: *Customer Booking Prediction for British Airways*
By: *Abu bakar*
Tools Used: Python · Scikit-learn · Pandas · Matplotlib · Imbalanced-learn
ML Techniques: Random Forest · Class Weights · SMOTE · ROC AUC · Pipelines

📞 *Questions? Let's talk!*

📧 *Armaan1900@outlook.com*