# 📊 Customer Churn Prediction Using Random Forest

## 1. What We Solved

We tackled the problem of predicting customer churn using the Telco Customer Churn dataset. Our goal was to classify whether a customer would churn (leave the service) or not, based on customer data like tenure, service type, contract, billing, and charges.

## 2. Dataset and Target

The dataset included 7043 customer records with a binary target variable: 'Churn' (Yes/No). Features included demographic data, service subscriptions, account information, and billing.

## 3. Initial Random Forest Model

We first trained a Random Forest model using default hyperparameters (only n\_estimators=100 defined). It gave us a baseline performance to compare future improvements.

## 4. Understanding Metrics

Accuracy: How many total predictions were correct.  
Precision: Of all churn predictions, how many were correct.  
Recall: Of all actual churners, how many did we catch.  
F1-Score: A balance between precision and recall.

## 5. First Output (Base Model)

The model had ~78.5% accuracy, but recall for churn (class 1) was just 0.47.  
It correctly predicted 175 churners but missed 199.  
Feature importance showed TotalCharges, MonthlyCharges, and Tenure were key predictors.

## 6. Hyperparameter Tuning

We used GridSearchCV to test 270 combinations of parameters:  
 - n\_estimators  
 - max\_depth  
 - min\_samples\_split  
 - min\_samples\_leaf  
 - max\_features  
Best Parameters Found:  
{'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 200}

## 7. Tuned Random Forest Results

Accuracy increased to ~79.6%. Churn recall improved slightly (0.48). F1-score improved to 0.56.

## 8. How Does the Model Know It's Wrong?

The model doesn’t know it’s wrong — we do.  
After it makes predictions on test data, we compare predictions to the actual Churn column.  
This lets us calculate accuracy, precision, recall, etc.

## 9. SMOTE – Balancing the Dataset

We used SMOTE to synthetically generate new examples of churners, balancing the training data from:  
Before SMOTE: 4130 non-churners vs. 1495 churners  
After SMOTE: 4130 churners and 4130 non-churners

## 10. Results After SMOTE + Tuned RF

Churn recall jumped to 0.68 (from 0.48), and F1-score improved to 0.60.  
This is a major win — we caught 255 churners (vs. 181 before).

## 11. Summary

- ✅ Baseline Random Forest: High accuracy, low recall for churn  
- ✅ Tuned RF: Small improvement in recall and F1  
- ✅ SMOTE + Tuned RF: Significant boost in recall, better balance  
  
Our final model (SMOTE + Tuned RF) is much better at identifying churners, which is the real business goal in churn prediction.