End-to-End ML Pipeline Report

1. Problem Statement & Objective

Problem: Predict customer churn for a telecommunications company.

Objective: Build an end-to-end machine learning pipeline that can preprocess data, train models, tune hyperparameters, and produce reusable pipelines for production.

2. Dataset Loading & Preprocessing

Dataset: Telco Customer Churn Dataset (WA_Fn-UseC_-Telco-Customer-Churn.csv).

Preprocessing steps performed:

- Handled missing values (TotalCharges and others).
- Removed duplicate rows.
- Split features into numeric and categorical.
- Built a ColumnTransformer for preprocessing:
  - Numeric: SimpleImputer (median) + StandardScaler.
  - Categorical: SimpleImputer (most frequent) + OneHotEncoder.
- Train-Test Split: 80%-20%.

3. Model Development & Training

Models used: Logistic Regression, Random Forest Classifier

Steps:

- Built pipelines combining preprocessing + model.
- Trained baseline models without hyperparameter tuning.
- Applied GridSearchCV for hyperparameter tuning:
  - Logistic Regression (10-fold CV)
  - Random Forest (5-fold CV)

4. Evaluation with Relevant Metrics

Metrics computed: Accuracy, ROC AUC, Precision, Recall, F1-Score for Churn=1 (target class).

Comparison performed:

- Untuned vs. tuned Logistic Regression
- Untuned vs. tuned Random Forest

Results (Logistic Regression tuned):

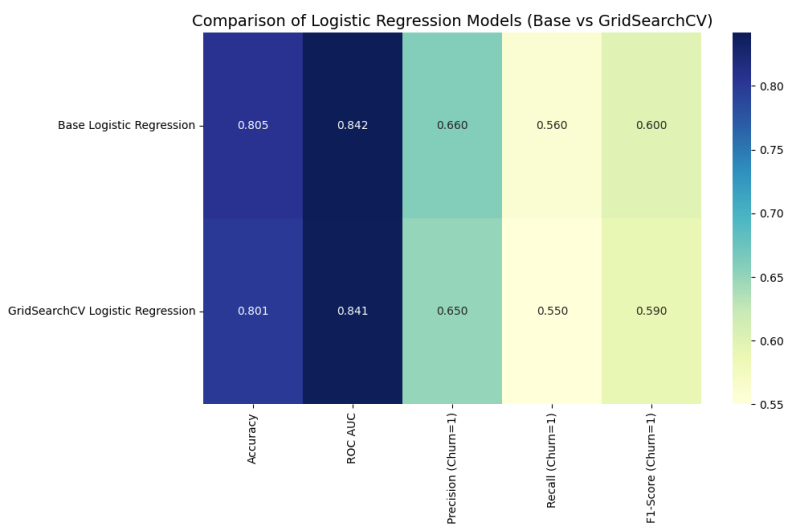
- Accuracy: 0.801

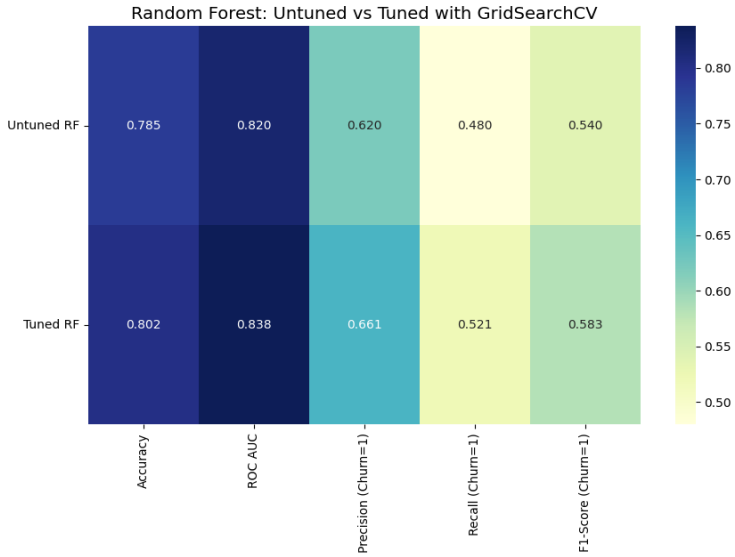
- ROC AUC: 0.841

- Precision (Churn=1): 0.650

- Recall (Churn=1): 0.550

5. Visualizations





6. Final Summary / Insights

- Preprocessing pipelines handle missing values, scaling, and encoding automatically.
- Hyperparameter tuning improves model performance or stabilizes metrics.
- Logistic Regression and Random Forest are both effective; Random Forest achieved slightly higher accuracy (0.802 vs 0.8006), while Logistic Regression had slightly better ROC AUC and recall for churned customers.
- Model selection depends on the priority: catching more churned customers (recall) or overall accuracy.
- Pipelines were exported using joblib, making them reusable and production-ready.
- The workflow covers end-to-end ML pipeline construction, evaluation, tuning, and deployment readiness.