# Telecom Customer Churn – Machine Learning Project

A practical end-to-end project that predicts customer churn for a telecom provider using classical ML models (Naive Bayes, K-Nearest Neighbors, Decision Tree). This README also includes a lightweight FastAPI inference service and Azure App Service (Docker) deployment steps.

## Features

• Data exploration & preprocessing (encoding, type fixes, drops)  
• Models: Naive Bayes, KNN, Decision Tree  
• Metrics: accuracy, precision, recall, F1  
• Reproducible training pipeline (scikit-learn)  
• FastAPI model server for real-time predictions  
• Containerized deployment to Azure via Docker + App Service

## Tech Stack

Python 3.10+, pandas, numpy, scikit-learn, matplotlib, seaborn, FastAPI, uvicorn, Docker, Azure CLI, Azure Container Registry (ACR), Azure App Service

## Project Structure

telecom-churn/  
├─ data/  
│ └─ telecom\_churn.csv  
├─ notebooks/  
│ └─ 01\_eda\_preprocess.ipynb  
├─ src/  
│ ├─ train.py  
│ ├─ preprocess.py  
│ ├─ model\_registry.py  
│ └─ schemas.py  
├─ api/  
│ └─ main.py  
├─ models/  
│ ├─ churn\_model.pkl  
│ └─ transformer.pkl  
├─ Dockerfile  
├─ requirements.txt  
└─ README.md

## Quickstart (Local)

1. Create environment & install dependencies  
2. Train models: python src/train.py --data data/telecom\_churn.csv --target Churn  
3. Run API locally: uvicorn api.main:app --host 0.0.0.0 --port 8000 --reload

## Training Details

• Drop non-predictive identifiers (customerID)  
• Convert data types and encode categorical variables  
• Map Churn labels: {No, Yes} → {0, 1}  
• Evaluate models using scikit-learn

## Azure Deployment

1. Login & create resource group: az group create -n rg-churn -l westeurope  
2. Create ACR and push Docker image  
3. Create Web App for Containers using az webapp create  
4. Configure settings and browse live API endpoint

## Requirements

fastapi, uvicorn, pandas, numpy, scikit-learn, joblib, matplotlib, seaborn

## License

MIT License