Customer Churn Prediction

🧠 **Project Assignment: Build & Deploy a Customer Churn Prediction Model**

Scenario:

You are working as a **Machine Learning Engineer** at a telecom company. Your manager has given you a task:
"Build a prediction model that can help us identify which customers are likely to leave our services."

Churn prediction is critical for businesses to retain customers. You'll use historical data to train a logistic regression model and deploy it as an interactive web app using **Streamlit**, so customer support or management can input a customer's details and instantly see the risk of churn.

Your Task:

You need to:

1. Understand and prepare the customer data
2. Train a Logistic Regression model to predict churn
3. Build a Streamlit web app for live predictions
4. Deploy the app online and share the link

Dataset Columns (Features Explained):

Here is the data you'll use:

- `gender`: Whether the customer is male or female
- `SeniorCitizen`: 1 if the customer is a senior citizen, 0 otherwise
- `Partner`: Does the customer have a partner? (Yes/No)
- `Dependents`: Does the customer have any dependents? (Yes/No)

- `tenure`: Number of months the customer has stayed with the company
- `PhoneService`: Does the customer have a phone service? (Yes/No)
- `MultipleLines`: Does the customer have multiple phone lines? (Yes/No/No phone service)
- `InternetService`: Customer's internet service provider (DSL/Fiber optic/No)
- `OnlineSecurity`: Does the customer have online security add-on? (Yes/No/No internet service)
- `OnlineBackup`: Does the customer have online backup add-on? (Yes/No/No internet service)
- `DeviceProtection`: Does the customer have device protection add-on? (Yes/No/No internet service)
- `TechSupport`: Does the customer have tech support? (Yes/No/No internet service)
- `StreamingTV`: Does the customer stream TV using the company's service? (Yes/No/No internet service)
- `StreamingMovies`: Does the customer stream movies using the company's service? (Yes/No/No internet service)
- `Contract`: Contract type – month-to-month, one year, or two year
- `PaperlessBilling`: Is billing done without paper? (Yes/No)
- `PaymentMethod`: How the customer pays (Credit card, bank transfer, etc.)
- `MonthlyCharges`: The amount charged to the customer monthly
- `TotalCharges`: Total amount charged over the entire tenure

- `Churn`: The target column – whether the customer left the service (Yes/No)

Step-by-Step Guide:

1. Data Loading and Exploration

- Load the dataset (CSV format)
- Explore the number of rows, column types, and any missing values
- Understand the balance of the target variable (`Churn`)

2. Preprocessing

- Convert `Churn` to binary: Yes → 1, No → 0
- Handle missing or blank values in `TotalCharges`
- Encode categorical columns using Label Encoding or One-Hot Encoding
- Normalize or scale `MonthlyCharges` and `TotalCharges` if needed
- Split the data into train and test sets (80:20 or 70:30)

3. Model Training

- Use **Logistic Regression** from `scikit-learn`
- Train the model on the training set
- Evaluate it on the test set using:
    - Accuracy
    - Precision, Recall, F1 Score

- Confusion Matrix

4. Streamlit Web App

Create a `app.py` file for Streamlit. It should:

- Allow users to input customer details (via dropdowns, sliders, etc.)
- Display prediction: "Customer is likely to churn" or "Customer is likely to stay"
- Optionally display model performance metrics or charts
- Have a clean and simple interface

5. Save & Deploy

- Save your model using `pickle` or `joblib`
- Create `requirements.txt` file
- Deploy on **Streamlit Cloud** (https://share.streamlit.io)
- Share the app link

Expected Deliverables:

1. Trained model (`logistic_model.pkl`)
2. Streamlit app (`app.py`)
3. GitHub repository with:
    - Code
    - README.md (project overview, how to run, model results)

    - requirements.txt
4. Live Streamlit App Link

End Result:

Your app should look something like this:

Input:

- Gender: Female
- Contract Type: Month-to-month
- Tenure: 2 months
- Monthly Charges: ₹70.5
  (...other fields filled in by the user)

Output:
🟡 **"This customer is likely to churn."**