Subscription Churn Predictor

A lightweight SaaS tool that predicts the likelihood of customer churn for subscription-based businesses. This tool helps businesses identify customers at risk and take proactive actions to retain them.

Features

- **Churn Prediction:** Enter customer data and get a probability score for their likelihood to churn.
- **Actionable Insights:** Receive suggestions on how to retain customers based on the churn probability.
- **User-Friendly Interface:** Simple form-based web app for input and results.
- **Lightweight ML Model:** Uses a basic decision-tree algorithm for predictions.

How It Works

1. Users input basic customer data (age, subscription duration, interaction frequency, and activity score) through a web form.
2. The system calculates the churn probability using a pre-trained Decision Tree Classifier.
3. The results page displays:
    - Churn probability percentage.
    - Suggested actions to retain the customer.

Tech Stack

- **Backend:** Flask (Python)
- **Frontend:** HTML/CSS (with optional Bootstrap for styling)
- **Machine Learning:** Scikit-learn
- **Database (optional):** SQLite

Setup Instructions

Follow these steps to set up the project on your local machine:

1. **Clone the Repository**
   bash
   Copy code
   git clone https://github.com/Dishaa-r/subscription-churn-predictor.git cd subscription-churn-predictor
2. **Install Dependencies** Use pip to install the required libraries:
   bash
   Copy code

pip install -r requirements.txt

3. **Run the Application** Start the Flask development server:
   bash
   Copy code
   python src/app.py
4. **Access the Web App** Open your browser and navigate to http://127.0.0.1:5000.

Project Structure

graphql

Copy code

- subscription-churn-predictor/ │
  - ├── README.md # Documentation
  - ├── requirements.txt # Dependencies
  - ├── src/ │
    - ├── app.py # Main Flask app │
    - ├── model.py # Churn prediction logic │
    - ├── utils.py # Helper functions (if any) │
  - ├── templates/ # HTML templates │
    - ├── index.html # Input form │
    - ├── results.html # Results display │
  - ├── static/ # Static files (CSS, images, etc.) │
    - ├── style.css # Styling for the app
  - │ └── tests/
    - ├── test_app.py # Unit tests

Dependencies

The project uses the following Python libraries:

- Flask
- Scikit-learn

Install them with:

bash

Copy code
pip install -r requirements.txt

Future Improvements

- Add support for bulk data uploads (e.g., via CSV).
- Allow users to save and retrieve past predictions.
- Enhance prediction accuracy with more advanced machine learning models.
- Add role-based authentication for multiple users.

License