**Customer churn prediction dashboard**using streamlit & gradio

**Workflow Summary**

1. 🔁 Code developed in Jupyter → pushed to **GitHub**
2. 🎯 Model trained → uploaded to **Hugging Face Model Hub**
3. 🎛️ Gradio interface built → deployed on **Hugging Face Spaces**
4. 🔗 GitHub README includes links to model and live demo

**🧩 You Can Replicate This Pattern**

For your **customer churn prediction dashboard**:

* Code + Streamlit app on GitHub
* Trained model on Hugging Face Model Hub (optional)
* Live demo on Hugging Face Spaces or Streamlit Cloud

Would you like help creating your GitHub + Hugging Face structure for your own project? I can scaffold it for you.

Great! Here's a ready-to-use **scaffold for your Customer Churn Prediction Dashboard** using both **GitHub** and **Hugging Face Spaces**, including a basic project structure, explanation, and a downloadable template (I’ll provide a .zip).

**🔧 Project Scaffold Overview**

customer-churn-dashboard/

├── app.py ← Streamlit app for dashboard (or Gradio if you prefer)

├── model/

│ └── model.pkl ← Trained ML model (e.g., Logistic Regression or XGBoost)

├── data/

│ └── sample.csv ← Sample input data

├── utils/

│ └── predict.py ← Python function to load model and predict

├── requirements.txt ← Required Python libraries

├── README.md ← Project description and instructions

├── .gitattributes ← Optional for large file support

└── HuggingFace README.md ← For model card if you upload model separately

**📁 1. GitHub Structure (Host your code)**

Use this to:

* Version your code
* Share training/inference scripts
* Collaborate
* Link to Hugging Face model/demo

You’ll push the entire above structure to your GitHub repo.

**🤗 2. Hugging Face Model Hub (Optional)**

* Upload your model.pkl and add a **model card** (README.md) with:

---

tags:

- churn-prediction

- tabular

- binary-classification

---

# Customer Churn Prediction Model

This model predicts whether a customer is likely to churn based on input features like tenure, charges, service history, etc.

Command to push:

transformers-cli login

git lfs install

git clone https://huggingface.co/your-username/churn-model

# Copy model.pkl and README.md inside the folder, then:

git add . && git commit -m "Add model" && git push

**🚀 3. Hugging Face Spaces (Live App)**

**Option 1: Streamlit App**

Use app.py:

import streamlit as st

import pandas as pd

import pickle

# Load model

with open("model/model.pkl", "rb") as f:

model = pickle.load(f)

st.title("Customer Churn Prediction")

input\_data = st.text\_area("Enter CSV-formatted customer data (1 row)")

if st.button("Predict"):

df = pd.read\_csv(pd.compat.StringIO(input\_data))

pred = model.predict(df)

st.write("Prediction:", "Churn" if pred[0] else "No Churn")

**Option 2: Gradio App**

Change app.py:

import gradio as gr

import pandas as pd

import pickle

model = pickle.load(open("model/model.pkl", "rb"))

def predict(csv\_string):

from io import StringIO

df = pd.read\_csv(StringIO(csv\_string))

result = model.predict(df)[0]

return "Churn" if result else "No Churn"

iface = gr.Interface(fn=predict, inputs="text", outputs="text")

iface.launch()

**📦 requirements.txt (common for both)**

streamlit==1.35.0

scikit-learn

pandas

numpy

gradio==4.29.0 # include if using Gradio

**🔗 README.md (in GitHub)**

# Customer Churn Prediction Dashboard

📊 Predict customer churn with a trained ML model.

## 🔧 How to Use

1. Clone this repo

2. Install dependencies: `pip install -r requirements.txt`

3. Run app: `streamlit run app.py` or `python app.py` (for Gradio)

## 🧠 Model Info

- Trained on XYZ dataset

- Accuracy: 88%

- Features: tenure, contract, monthly charges, etc.

👉 Try the live app: [Hugging Face Space](https://huggingface.co/spaces/your-name/churn-predictor)

👉 Model on Hugging Face: [Model Link](https://huggingface.co/your-name/churn-model)

**🎁 Want a Downloadable Template?**

I can now generate and send you a ready-to-edit .zip file containing:

* All folder structure
* Sample model.pkl (dummy)
* Basic Streamlit app
* README

**Shall I send it now?** Just confirm and I’ll upload it here for you.

Your project scaffold is ready! You can download the .zip file below:

👉 [Download customer-churn-dashboard.zip](sandbox://mnt/data/customer-churn-dashboard.zip)

This includes:

* A dummy model.pkl
* A sample Streamlit app
* Predict utility code
* Sample input CSV
* All ready to test locally or deploy on GitHub + Hugging Face Spaces.

Let me know if you want the same setup for a **Gradio app** or help pushing this to GitHub/Hugging Face.

It seems like I can’t do more advanced data analysis right now. Please try again later.

In the meantime, here’s what you can do manually:

**✅ 1. For Gradio App**

Add this to your project folder as app\_gradio.py:

import gradio as gr

import pandas as pd

import pickle

# Load model

with open("model/model.pkl", "rb") as f:

model = pickle.load(f)

def predict\_churn(csv\_string):

try:

from io import StringIO

df = pd.read\_csv(StringIO(csv\_string))

prediction = model(df)

return "Churn" if prediction[0] else "No Churn"

except Exception as e:

return f"Error: {str(e)}"

iface = gr.Interface(

fn=predict\_churn,

inputs="text",

outputs="text",

title="Customer Churn Prediction",

description="Paste a single row of CSV-formatted customer data to predict churn."

)

if \_\_name\_\_ == "\_\_main\_\_":

iface.launch()

**📦 Update your requirements.txt:**

streamlit==1.35.0

pandas

numpy

scikit-learn

gradio==4.29.0

**📤 2. Push to GitHub:**

1. Go to <https://github.com/new> and create a new repo (e.g., churn-dashboard)
2. In your terminal:
3. cd path/to/customer-churn-dashboard
4. git init
5. git remote add origin https://github.com/YOUR\_USERNAME/churn-dashboard.git
6. git add .
7. git commit -m "Initial commit"
8. git push -u origin main

**🤗 3. Deploy on Hugging Face Spaces:**

1. Go to <https://huggingface.co/spaces>
2. Click **"Create new Space"**
3. Choose:
   * **SDK**: Streamlit or Gradio (depending on app.py or app\_gradio.py)
   * **Repo name**: churn-predictor
4. Clone the repo locally and push:
5. git clone https://huggingface.co/spaces/YOUR\_USERNAME/churn-predictor
6. cd churn-predictor
7. cp -r path/to/customer-churn-dashboard/\* .
8. git add .
9. git commit -m "Add churn dashboard"
10. git push

**For gradio app  
  
create it manually by:**

1. **Using the earlier zip you downloaded.**
2. **Adding a new file named app\_gradio.py with the Gradio code I shared earlier.**
3. **Updating requirements.txt to include gradio==4.29.0.**