### import streamlit as st

### import pandas as pd

### import numpy as np

### import matplotlib.pyplot as plt

### import seaborn as sns

### from sklearn.model\_selection import train\_test\_split

### from sklearn.preprocessing import LabelEncoder

### from sklearn.linear\_model import LogisticRegression

### from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier

### from xgboost import XGBClassifier

### from lightgbm import LGBMClassifier

### from sklearn.metrics import precision\_score, recall\_score, log\_loss

### import warnings

### warnings.filterwarnings('ignore')

### # ---------------- Page Settings ----------------

### st.set\_page\_config(page\_title="Microfinance Loan Predictor", layout="wide")

### # ---------------- Custom CSS ----------------

### st.markdown("""

### <style>

### body {background-color: #f9fafc;}

### .main-title {text-align: center; color: #2e86de; font-size: 38px; font-weight: bold;}

### .sub-title {text-align: center; color: #555; font-size: 18px; margin-bottom: 20px;}

### .footer {text-align: center; font-size: 14px; color: #888; margin-top: 40px;}

### .section-title {color: #2e86de; font-size: 24px; font-weight: bold; margin-top: 20px;}

### </style>

### """, unsafe\_allow\_html=True)

### # ---------------- Sidebar ----------------

### menu = ["Home", "About", "Prediction", "Results", "Contact"]

### choice = st.sidebar.radio(" Navigate", menu)

### # ---------------- Dataset Loading ----------------

### file\_path = "D:/microfinance\_app/Micro-credit-Data-file.csv"

### try:

### df = pd.read\_csv(file\_path)

### df.columns = df.columns.str.replace('[^A-Za-z0-9\_]+', '\_', regex=True)

### except FileNotFoundError:

### df = None

### st.error(" Dataset file not found. Please check the path.")

### # ---------------- Home Page ----------------

### if choice == "Home":

### st.markdown("<h1 class='main-title'>Microfinance Loan Repayment Predictor</h1>", unsafe\_allow\_html=True)

### st.markdown("<p class='sub-title'>Empowering microfinance institutions with predictive insights for smarter lending.</p>", unsafe\_allow\_html=True)

### st.image("D:/microfinance\_app/home.jpg", use\_container\_width=False, width=600)

### st.write("""

### Welcome to the Microfinance Loan Repayment Predictor!

### This platform helps financial institutions assess the likelihood of borrowers repaying their micro-loans on time.

### 

### ### Features:

### - Interactive data visualization

### - AI-based loan repayment prediction

### - Model comparison for best performance

### - Real-time dataset insights

### Microfinance plays a crucial role in empowering low-income individuals. This app supports \*\*data-driven decision-making\*\* to minimize defaults and maximize impact.

### """)

### # ---------------- About Page ----------------

### elif choice == "About":

### st.header("ℹ️ About the Project")

### st.image("D:/microfinance\_app/about.jpg", use\_container\_width=False, width=550)

### st.write("""

### Microfinance institutions aim to provide financial access to individuals who may not qualify for traditional bank loans.

### However, ensuring timely repayment remains a major challenge.

### ### Project Objective:

### To build a machine learning model that predicts whether a borrower is likely to repay their loan within 5 days.

### 

### ### Technologies Used:

### - Python

### - Streamlit for web deployment

### - Machine Learning models: Logistic Regression, Random Forest, Gradient Boosting, XGBoost, LightGBM

### - Data Visualization with Seaborn & Matplotlib

### This project showcases how data analytics can enhance microfinance sustainability and decision-making.

### """)

### # ---------------- Prediction Page ----------------

### elif choice == "Prediction":

### st.header(" Loan Repayment Prediction")

### st.write("Fill in borrower details to check repayment likelihood:")

### col1, col2 = st.columns(2)

### with col1:

### age = st.number\_input("Age", 18, 70, 30)

### income = st.number\_input("Monthly Income", 1000, 100000, 5000)

### with col2:

### loan\_amount = st.number\_input("Loan Amount", 1000, 50000, 10000)

### credit\_history = st.selectbox("Credit History", ["Good", "Bad"])

### if st.button("Predict"):

### if (income > loan\_amount) and (credit\_history == "Good"):

### st.success("Likely to Repay (Confidence ~80%)")

### elif (income > loan\_amount):

### st.info("Medium Risk but Possibly Repay (Confidence ~60%)")

### else:

### st.error(" High Risk of Default (Confidence ~70%)")

### # ---------------- Results Page ----------------

### elif choice == "Results":

### st.header(" Model Results & Data Insights")

### if df is not None:

### # Dataset Preview

### st.subheader(" Dataset Overview")

### st.dataframe(df.head())

### # Target Distribution

### if 'label' in df.columns:

### st.subheader("Distribution of Target Variable")

### fig, ax = plt.subplots()

### sns.countplot(x='label', data=df, palette='coolwarm', ax=ax)

### ax.set\_title('Distribution of Target Variable (Defaulters vs Non-Defaulters)')

### st.pyplot(fig)

### # Correlation Heatmap

### st.subheader(" Feature Correlation Heatmap")

### fig, ax = plt.subplots(figsize=(10, 6))

### sns.heatmap(df.select\_dtypes(include=[np.number]).corr(), cmap='coolwarm', ax=ax)

### st.pyplot(fig)

### # Model Training Section

### st.subheader(" Model Performance Comparison")

### # Encode and split data

### cat\_cols = df.select\_dtypes(include='object').columns

### for col in cat\_cols:

### le = LabelEncoder()

### df[col] = le.fit\_transform(df[col].astype(str))

### X = df.drop(['label'], axis=1)

### y = df['label']

### X\_train, X\_valid, y\_train, y\_valid = train\_test\_split(X, y, test\_size=0.2, stratify=y, random\_state=42)

### models = {

### 'LogisticRegression': LogisticRegression(max\_iter=500),

### 'RandomForest': RandomForestClassifier(),

### 'GradientBoosting': GradientBoostingClassifier(),

### 'XGBoost': XGBClassifier(eval\_metric='logloss'),

### 'LightGBM': LGBMClassifier()

### }

### results = []

### for name, model in models.items():

### model.fit(X\_train, y\_train)

### y\_pred = model.predict(X\_valid)

### y\_proba = model.predict\_proba(X\_valid)[:, 1]

### precision = precision\_score(y\_valid, y\_pred)

### recall = recall\_score(y\_valid, y\_pred)

### loss = log\_loss(y\_valid, y\_proba)

### results.append([name, precision, recall, loss])

### results\_df = pd.DataFrame(results, columns=['Model', 'Precision', 'Recall', 'Log Loss'])

### st.dataframe(results\_df)

### else:

### st.warning(" Dataset not loaded. Please check the file path.")

### # ---------------- Contact Page ----------------

### elif choice == "Contact":

### st.header(" Contact Us")

### st.image("D:/microfinance\_app/result.jpg", use\_container\_width=False, width=550)

### 

### st.write("""

### ### Project Owners:

### PRAISY VICTOR

### Email: praisyvictor20@gmail.com

### GitHub: [Praisy's GitHub](https://github.com/)

### LinkedIn: [Praisy's LinkedIn](https://linkedin.com/)

### POOJA SHREE

### Email: shreekarthikeyan06@gmail.com

### GitHub: [Pooja's GitHub](https://github.com/)

### LinkedIn: [Pooja's LinkedIn](https://linkedin.com/)

### ---

### \*This project demonstrates how data analytics and AI can revolutionize microfinance.\*

### """)

### st.markdown("<p class='footer'>© 2025 Microfinance Loan Repayment Predictor | Built with Streamlit</p>", unsafe\_allow\_html=True)