

## app8.py

```
1 # loan_eligibility_app.py
2
3 import streamlit as st
4 import pandas as pd
5 import numpy as np
6 from sklearn.linear_model import LogisticRegression
7 from sklearn.model_selection import train_test_split
8 from sklearn.preprocessing import LabelEncoder
9
10 st.set_page_config(page_title="Loan Eligibility Predictor", layout="centered")
11 st.title("🏠 Loan Eligibility Prediction App")
12
13 # Load and preprocess dataset
14 @st.cache_data
15 def load_and_train_model():
16     df = pd.read_csv("loan_train.csv")
17
18     # Fill missing values
19     for col in ['Gender', 'Married', 'Dependents', 'Self_Employed', 'Loan_Amount_Term',
20 'Credit_History']:
21         df[col].fillna(df[col].mode()[0], inplace=True)
22     df['LoanAmount'].fillna(df['LoanAmount'].mean(), inplace=True)
23
24     # Feature engineering
25     df['LoanAmount_log'] = np.log(df['LoanAmount'])
26     df['TotalIncome'] = df['ApplicantIncome'] + df['CoapplicantIncome']
27     df['TotalIncome_log'] = np.log(df['TotalIncome'])
28
29     # Label encoding
30     cols = ['Gender', 'Married', 'Education', 'Self_Employed', 'Property_Area',
31 'Loan_Status', 'Dependents']
32     for col in cols:
33         df[col] = LabelEncoder().fit_transform(df[col])
34
35     X = df[['Credit_History', 'Education', 'Married', 'Self_Employed', 'TotalIncome_log',
36 'LoanAmount_log']]
37     y = df['Loan_Status']
38
39     model = LogisticRegression()
40     model.fit(X, y)
41     return model
42
43 model = load_and_train_model()
44
45 # Sidebar Inputs
46 st.sidebar.header("Applicant Information")
47 gender = st.sidebar.selectbox("Gender", ['Male', 'Female'])
48 married = st.sidebar.selectbox("Married", ['Yes', 'No'])
49 education = st.sidebar.selectbox("Education", ['Graduate', 'Not Graduate'])
50 self_employed = st.sidebar.selectbox("Self Employed", ['Yes', 'No'])
51 credit_history = st.sidebar.selectbox("Credit History", [1.0, 0.0])
52 applicant_income = st.sidebar.number_input("Applicant Income", 0, 100000, 5000)
```

```
50 coapplicant_income = st.sidebar.number_input("Coapplicant Income", 0, 100000, 0)
51 loan_amount = st.sidebar.number_input("Loan Amount (in thousands)", 1, 1000, 200)
52
53 # Feature transformation
54 total_income = applicant_income + coapplicant_income
55 total_income_log = np.log(total_income) if total_income > 0 else 0
56 loan_amount_log = np.log(loan_amount) if loan_amount > 0 else 0
57
58 # Manual encoding
59 gender = 1 if gender == 'Male' else 0
60 married = 1 if married == 'Yes' else 0
61 education = 1 if education == 'Graduate' else 0
62 self_employed = 1 if self_employed == 'Yes' else 0
63
64 # Prediction
65 input_data = np.array([[credit_history, education, married, self_employed,
66 total_income_log, loan_amount_log]])
67 prediction = model.predict(input_data)[0]
68
69 # Output
70 st.subheader("Prediction Result:")
71 if prediction == 1:
72     st.success("✅ Loan is likely to be Approved")
73 else:
74     st.error("❌ Loan is likely to be Rejected")
```