```
#Import libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


#Load dataset (change path if needed)
df = pd.read_csv("/content/credit_risk_dataset.csv")


#Basic info
print("Shape:", df.shape)
print(df.head())
print(df.info())
print(df['loan_status'].value_counts())   # target column
```

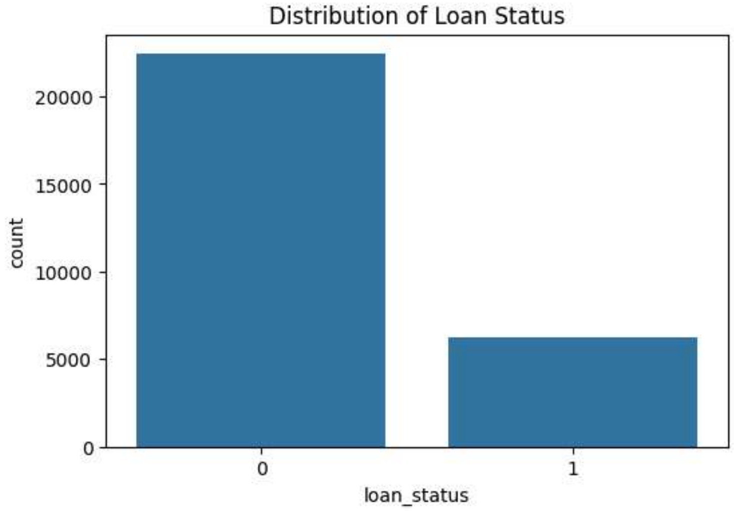
```
Shape: (32581, 12)
   person_age  person_income person_home_ownership  person_emp_length  \
0          22          59000                  RENT              123.0
1          21           9600                   OWN                5.0
2          25           9600              MORTGAGE                1.0
3          23          65500                  RENT                4.0
4          24          54400                  RENT                8.0

  loan_intent loan_grade  loan_amnt  loan_int_rate  loan_status  \
0    PERSONAL          D      35000          16.02            1
1   EDUCATION          B       1000          11.14            0
2     MEDICAL          C       5500          12.87            1
3     MEDICAL          C      35000          15.23            1
4     MEDICAL          C      35000          14.27            1

   loan_percent_income cb_person_default_on_file  cb_person_cred_hist_length
0                 0.59                         Y                           3
1                 0.10                         N                           2
2                 0.57                         N                           3
3                 0.53                         N                           2
4                 0.55                         Y                           4
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 32581 entries, 0 to 32580
Data columns (total 12 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   person_age                  32581 non-null  int64  
 1   person_income               32581 non-null  int64  
 2   person_home_ownership       32581 non-null  object 
 3   person_emp_length           31686 non-null  float64
 4   loan_intent                 32581 non-null  object 
 5   loan_grade                  32581 non-null  object 
 6   loan_amnt                   32581 non-null  int64  
 7   loan_int_rate               29465 non-null  float64
 8   loan_status                 32581 non-null  int64  
 9   loan_percent_income         32581 non-null  float64
 10  cb_person_default_on_file   32581 non-null  object 
 11  cb_person_cred_hist_length  32581 non-null  int64  
dtypes: float64(3), int64(5), object(4)
memory usage: 3.0+ MB
None
loan_status
0    25473
1     7108
Name: count, dtype: int64
```

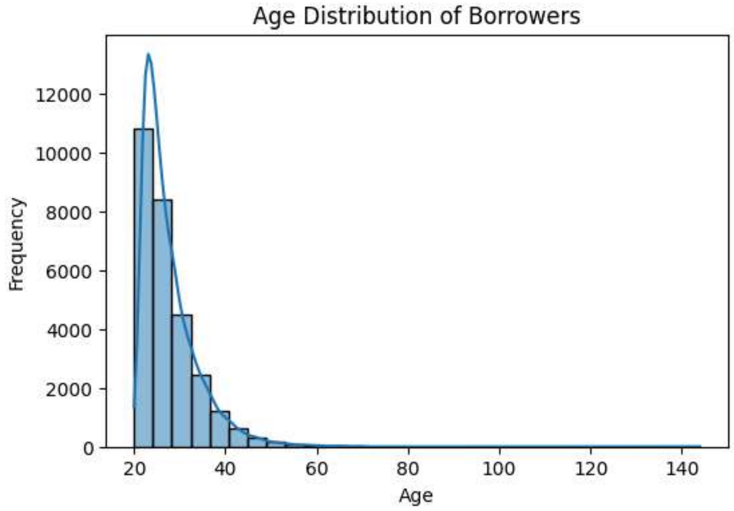
```
#Check missing values
print(df.isnull().sum())

#Drop rows with missing data (if very few)
df = df.dropna()

#Convert categorical variables if needed
cat_cols = df.select_dtypes(include=['object']).columns
print("Categorical columns:", cat_cols)

#Example: Encode categorical features
df = pd.get_dummies(df, drop_first=True)
```

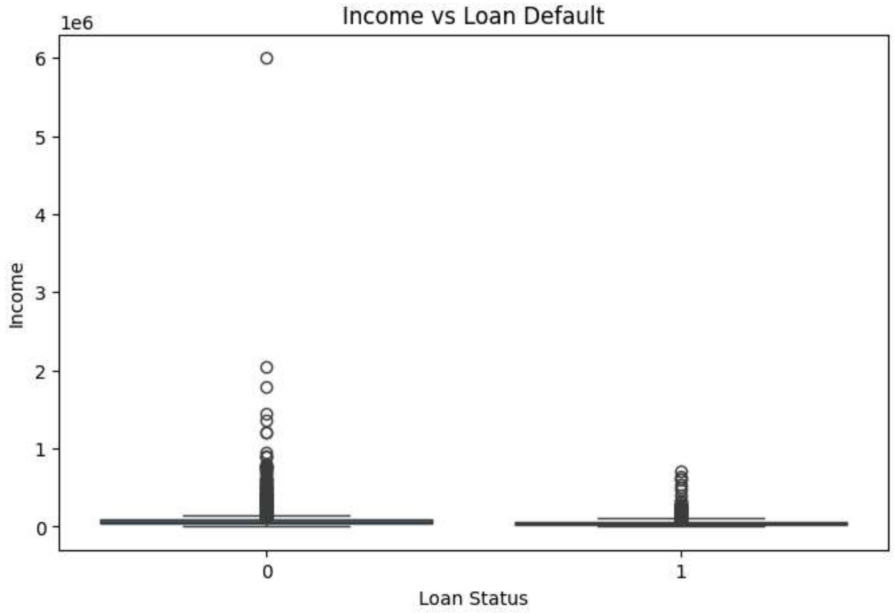
```
person_age                       0
person_income                    0
person_home_ownership            0
person_emp_length              895
loan_intent                      0
loan_grade                       0
loan_amnt                        0
loan_int_rate                 3116
loan_status                      0
loan_percent_income              0
```

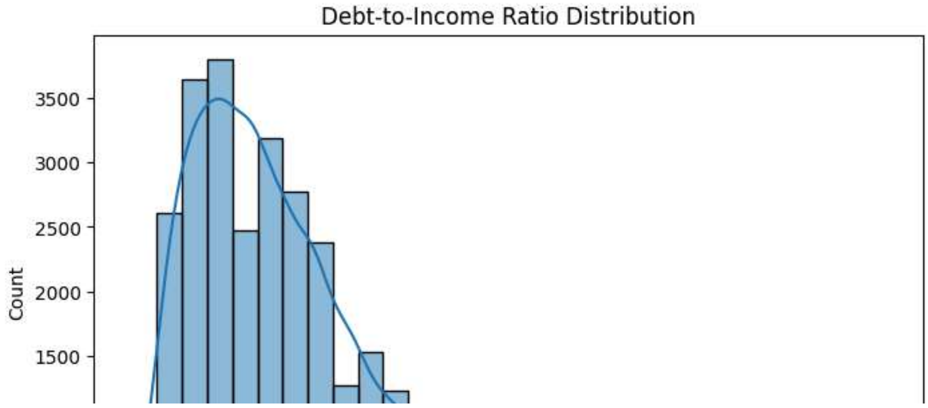
```
cb_person_default_on_file        0
cb_person_cred_hist_length       0
dtype: int64
Categorical columns: Index(['person_home_ownership', 'loan_intent', 'loan_grade',
       'cb_person_default_on_file'],
      dtype='object')
```

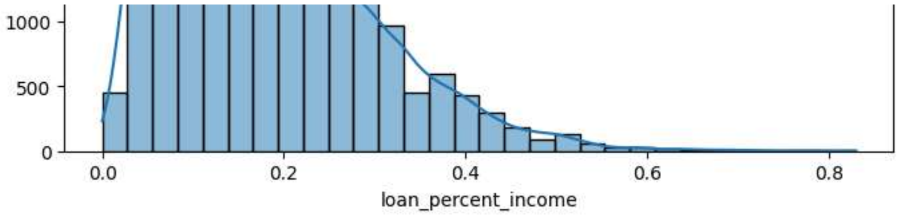
```
#Distribution of Loan Status (Target)
plt.figure(figsize=(6, 4))
sns.countplot(x="loan_status", data=df)
plt.title("Distribution of Loan Status")
plt.show()

#Age Distribution
plt.figure(figsize=(6, 4))
sns.histplot(df['person_age'], bins=30, kde=True)
plt.title("Age Distribution of Borrowers")
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.show()

#Income vs Loan Default
plt.figure(figsize=(8,5))
sns.boxplot(x="loan_status", y="person_income", data=df)
plt.title("Income vs Loan Default")
plt.xlabel("Loan Status")
plt.ylabel("Income")
plt.show()

#Debt-to-Income Ratio Distribution
plt.figure(figsize=(8, 5))
sns.histplot(df['loan_percent_income'], bins=30, kde=True)
plt.title("Debt-to-Income Ratio Distribution")
plt.show()
```


Distribution of Loan Status
20000
15000
10000
5000
0
count
0
1
loan_status

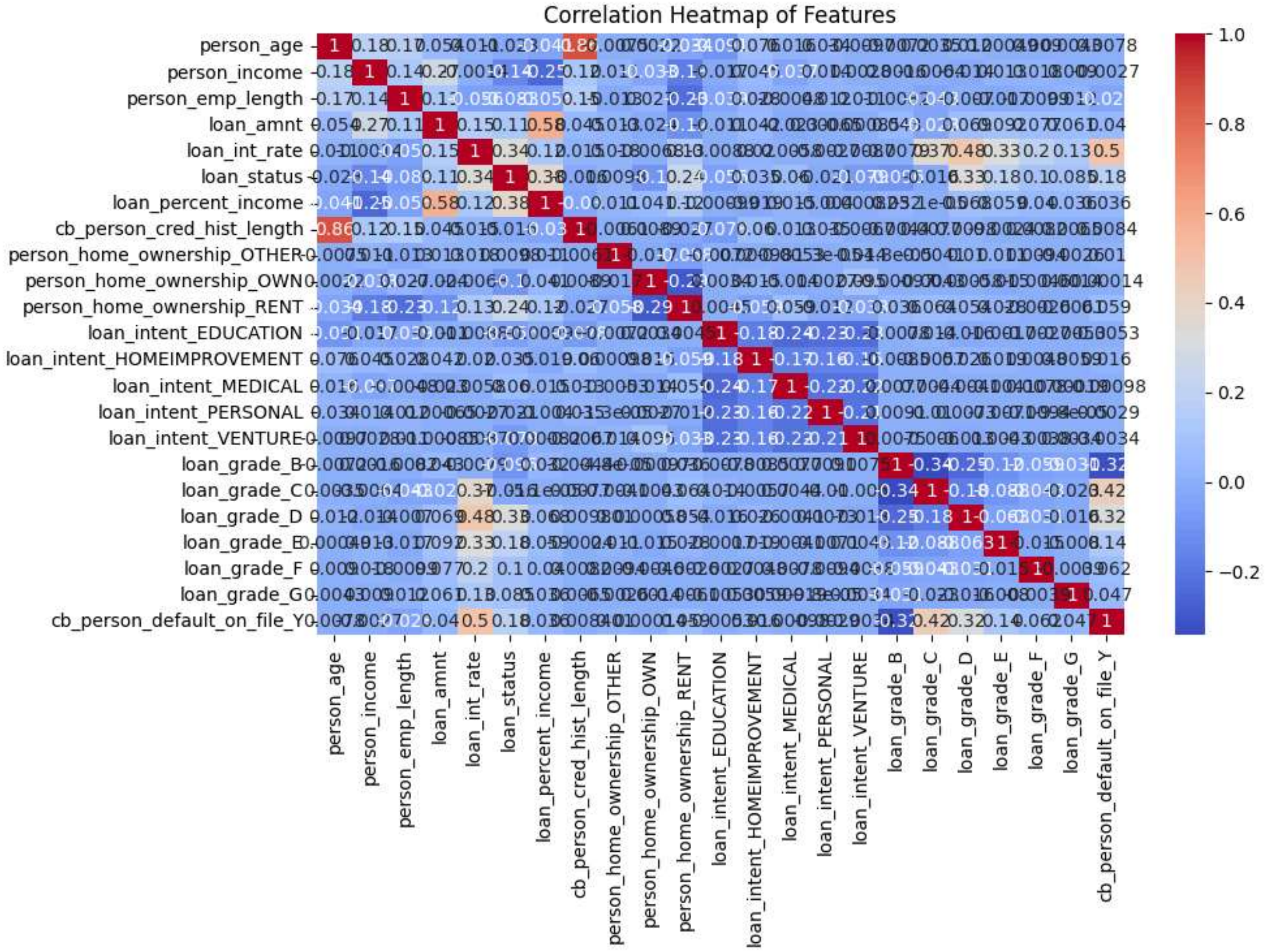

Age Distribution of Borrowers
12000
10000
8000
6000
4000
2000
0
Frequency
20
40
60
80
100
120
140
Age


Income vs Loan Default
1e6
6
5
4
3
2
1
0
Income
0
1
Loan Status


Debt-to-Income Ratio Distribution
3500
3000
2500
2000
1500
Count



```
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(numeric_only=True), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap of Features")
plt.show()
```



```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix

# Features & Target
X = df.drop("loan_status", axis=1)
y = df["loan_status"]
```