

```
In [1]: import pandas as pd

# Load the dataset
data = pd.read_csv('loan_detection.csv')

In [3]: # Check the first few rows of the dataset
print(data.head())

# Check for missing values
print(data.isnull().sum())

age      campaign      pdays      previous      no_previous_contact      not_working \
0      56      1      999      0      1      0
1      57      1      999      0      1      0
2      37      1      999      0      1      0
3      40      1      999      0      1      0
4      56      1      999      0      1      0

job_admin      job_blue-collar      job_entrepreneur      job_housemaid      ... \
0      0      0      0      0      1 ...
1      0      0      0      0      0 ...
2      0      0      0      0      0 ...
3      1      0      0      0      0 ...
4      0      0      0      0      0 ...

month_sep      day_of_week_fri      day_of_week_mon      day_of_week_thu \
0      0      0      0      1      0
1      0      0      0      1      0
2      0      0      0      1      0
3      0      0      0      1      0
4      0      0      0      1      0

day_of_week_tue      day_of_week_wed      poutcome_failure      poutcome_nonexistent \
0      0      0      0      0      1
1      0      0      0      0      1
2      0      0      0      0      1
3      0      0      0      0      1
4      0      0      0      0      1

poutcome_success      Loan_Status_label
0      0      0
1      0      0
2      0      0
3      0      0
4      0      0

[5 rows x 60 columns]

age      0
campaign      0
pdays      0
previous      0
no_previous_contact      0
not_working      0
job_admin      0
job_blue-collar      0
job_entrepreneur      0
job_housemaid      0
job_management      0
job_retired      0
job_self-employed      0
job_services      0
job_student      0
job_technician      0
job_unemployed      0
job_unknown      0
marital_divorced      0
marital_married      0
marital_single      0
marital_unknown      0
education_basic_4y      0
education_basic_6y      0
education_basic_8y      0
education_high_school      0
education_illiterate      0
education_professional_course      0
education_university_degree      0
education_unknown      0
default_no      0
default_unknown      0
default_yes      0
housing_no      0
housing_unknown      0
housing_yes      0
loan_no      0
loan_unknown      0
loan_yes      0
contact_cellular      0
contact_telephone      0
month_apr      0
month_aug      0
month_dec      0
month_jul      0
month_jun      0
month_may      0
month_nov      0
month_oct      0
month_sep      0
day_of_week_fri      0
day_of_week_mon      0
day_of_week_thu      0
day_of_week_tue      0
day_of_week_wed      0
poutcome_failure      0
poutcome_nonexistent      0
poutcome_success      0
Loan_Status_label      0
dtype: int64

In [4]: # One-hot encoding for categorical features
data = pd.get_dummies(data, drop_first=True)

In [5]: X = data.drop('Loan_Status_label', axis=1)
y = data['Loan_Status_label']

In [7]: from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

In [8]: from sklearn.ensemble import RandomForestClassifier

# Initialize the model
model = RandomForestClassifier(random_state=42)

# Train the model
model.fit(X_train, y_train)

Out[8]:
RandomForestClassifier
RandomForestClassifier(random_state=42)

In [9]: y_pred = model.predict(X_test)

In [10]: from sklearn.metrics import classification_report, confusion_matrix

# Print classification report
print(classification_report(y_test, y_pred))

# Print confusion matrix
print(confusion_matrix(y_test, y_pred))

              precision    recall  f1-score   support

      0       0.91      0.97      0.94       7303
      1       0.51      0.25      0.33        935

 accuracy      0.71      0.61      0.64      8238
macro avg      0.71      0.61      0.64      8238
weighted avg      0.86      0.89      0.87      8238

[[7674 229]
 [ 701 234]]

In [14]: import matplotlib.pyplot as plt
import numpy as np

# Get feature importances
importances = model.feature_importances_

# Sort features by importance
indices = np.argsort(importances)[::-1]

# Plot the feature importances
plt.figure()
plt.title("Feature Importances")
plt.bar(range(X.shape[1]), importances[indices], align="center")
plt.xticks(range(X.shape[1]), X.columns[indices], rotation=90)
plt.xlim([-1, X.shape[1]])
plt.show()

Feature Importances

0.20
0.15
0.10
0.05
0.00

age
campaign
pdays
previous
no_previous_contact
not_working
job_admin
job_blue-collar
job_entrepreneur
job_housemaid
job_management
job_retired
job_self-employed
job_services
job_student
job_technician
job_unemployed
job_unknown
marital_divorced
marital_married
marital_single
marital_unknown
education_basic_4y
education_basic_6y
education_basic_8y
education_high_school
education_illiterate
education_professional_course
education_university_degree
education_unknown
default_no
default_unknown
default_yes
housing_no
housing_unknown
housing_yes
loan_no
loan_unknown
loan_yes
contact_cellular
contact_telephone
month_apr
month_aug
month_dec
month_jul
month_jun
month_may
month_nov
month_oct
month_sep
day_of_week_fri
day_of_week_mon
day_of_week_thu
day_of_week_tue
day_of_week_wed
poutcome_failure
poutcome_nonexistent
poutcome_success
Loan_Status_label
dtype: object

In [15]: import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
import matplotlib.pyplot as plt

# Load the dataset
data = pd.read_csv('loan_detection.csv')

# Data preprocessing
data = pd.get_dummies(data, drop_first=True)

# Define features and target variable
X = data.drop('Loan_Status_label', axis=1)
y = data['Loan_Status_label']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize and train the model
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

# Feature importance
importances = model.feature_importances_
indices = np.argsort(importances)[::-1]

# Plot feature importances
plt.figure()
plt.title("Feature Importances")
plt.bar(range(X.shape[1]), importances[indices], align="center")
plt.xticks(range(X.shape[1]), X.columns[indices], rotation=90)
plt.xlim([-1, X.shape[1]])
plt.show()

              precision    recall  f1-score   support

      0       0.91      0.97      0.94       7303
      1       0.51      0.25      0.33        935

 accuracy      0.71      0.61      0.64      8238
macro avg      0.71      0.61      0.64      8238
weighted avg      0.86      0.89      0.87      8238

[[7674 229]
 [ 701 234]]

Feature Importances

0.20
0.15
0.10
0.05
0.00

age
campaign
pdays
previous
no_previous_contact
not_working
job_admin
job_blue-collar
job_entrepreneur
job_housemaid
job_management
job_retired
job_self-employed
job_services
job_student
job_technician
job_unemployed
job_unknown
marital_divorced
marital_married
marital_single
marital_unknown
education_basic_4y
education_basic_6y
education_basic_8y
education_high_school
education_illiterate
education_professional_course
education_university_degree
education_unknown
default_no
default_unknown
default_yes
housing_no
housing_unknown
housing_yes
loan_no
loan_unknown
loan_yes
contact_cellular
contact_telephone
month_apr
month_aug
month_dec
month_jul
month_jun
month_may
month_nov
month_oct
month_sep
day_of_week_fri
day_of_week_mon
day_of_week_thu
day_of_week_tue
day_of_week_wed
poutcome_failure
poutcome_nonexistent
poutcome_success
Loan_Status_label
dtype: object

In [16]: # Display the first few rows
print(data.head())

# Check the data types and for missing values
print(data.info())
print(data.isnull().sum())

age      campaign      pdays      previous      no_previous_contact      not_working \
0      56      1      999      0      1      0
1      57      1      999      0      1      0
2      37      1      999      0      1      0
3      40      1      999      0      1      0
4      56      1      999      0      1      0

job_admin      job_blue-collar      job_entrepreneur      job_housemaid      ... \
0      0      0      0      0      1 ...
1      0      0      0      0      0 ...
2      0      0      0      0      0 ...
3      1      0      0      0      0 ...
4      0      0      0      0      0 ...

month_sep      day_of_week_fri      day_of_week_mon      day_of_week_thu \
0      0      0      0      1      0
1      0      0      0      1      0
2      0      0      0      1      0
3      0      0      0      1      0
4      0      0      0      1      0

day_of_week_tue      day_of_week_wed      poutcome_failure      poutcome_nonexistent \
0      0      0      0      0      1
1      0      0      0      0      1
2      0      0      0      0      1
3      0      0      0      0      1
4      0      0      0      0      1

poutcome_success      Loan_Status_label
0      0      0
1      0      0
2      0      0
3      0      0
4      0      0

[5 rows x 60 columns]

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41188 entries, 0 to 41187
Data columns (total 60 columns):
#   Column                Non-Null Count  Dtype
---  --
0   age                   41188 non-null    int64
1   campaign              41188 non-null    int64
2   pdays                 41188 non-null    int64
3   previous              41188 non-null    int64
4   no_previous_contact    41188 non-null    int64
5   not_working           41188 non-null    int64
6   job_admin             41188 non-null    int64
7   job_blue-collar       41188 non-null    int64
8   job_entrepreneur      41188 non-null    int64
9   job_housemaid         41188 non-null    int64
10  job_management        41188 non-null    int64
11  job_retired           41188 non-null    int64
12  job_self-employed     41188 non-null    int64
13  job_services          41188 non-null    int64
14  job_student           41188 non-null    int64
15  job_technician        41188 non-null    int64
16  job_unemployed        41188 non-null    int64
17  job_unknown           41188 non-null    int64
18  marital_divorced      41188 non-null    int64
19  marital_married       41188 non-null    int64
20  marital_single        41188 non-null    int64
21  marital_unknown       41188 non-null    int64
22  education_basic_4y    41188 non-null    int64
23  education_basic_6y    41188 non-null    int64
24  education_basic_8y    41188 non-null    int64
25  education_high_school 41188 non-null    int64
26  education_illiterate  41188 non-null    int64
27  education_professional_course 41188 non-null    int64
28  education_university_degree 41188 non-null    int64
29  education_unknown    41188 non-null    int64
30  default_no           41188 non-null    int64
31  default_unknown      41188 non-null    int64
32  default_yes          41188 non-null    int64
33  housing_no           41188 non-null    int64
34  housing_unknown      41188 non-null    int64
35  housing_yes          41188 non-null    int64
36  loan_no              41188 non-null    int64
37  loan_unknown         41188 non-null    int64
38  loan_yes             41188 non-null    int64
39  contact_cellular     41188 non-null    int64
40  contact_telephone    41188 non-null    int64
41  month_apr            41188 non-null    int64
42  month_aug            41188 non-null    int64
43  month_dec            41188 non-null    int64
44  month_jul            41188 non-null    int64
45  month_jun            41188 non-null    int64
46  month_may            41188 non-null    int64
47  month_nov            41188 non-null    int64
48  month_oct            41188 non-null    int64
49
```


