

```

In [1]: import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.metrics import classification_report, accuracy_score

# Load the dataset
df = pd.read_csv('train_u6lujuX_CVtuZ9i (1).csv')

# Drop Loan_ID as it's not useful for prediction
df.drop('Loan_ID', axis=1, inplace=True)

# Separate target variable
y = df['Loan_Status']
X = df.drop('Loan_Status', axis=1)

# Encode target variable
y = y.map({'Y': 1, 'N': 0})

# Encode categorical features
categorical_cols = X.select_dtypes(include='object').columns
X[categorical_cols] = X[categorical_cols].apply(LabelEncoder().fit_transform)

# Impute missing values
imputer = SimpleImputer(strategy='mean')
X = pd.DataFrame(imputer.fit_transform(X), columns=X.columns)

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

# Feature scaling
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Logistic Regression
lr = LogisticRegression()
lr.fit(X_train, y_train)
lr_preds = lr.predict(X_test)
print("Logistic Regression Accuracy:", accuracy_score(y_test, lr_preds))
print(classification_report(y_test, lr_preds))

# Support Vector Machine
svm = SVC(kernel='linear')
svm.fit(X_train, y_train)
svm_preds = svm.predict(X_test)
print("SVM Accuracy:", accuracy_score(y_test, svm_preds))
print(classification_report(y_test, svm_preds))

```

Logistic Regression Accuracy: 0.7886178861788617

	precision	recall	f1-score	support
0	0.95	0.42	0.58	43
1	0.76	0.99	0.86	80
accuracy			0.79	123
macro avg	0.85	0.70	0.72	123
weighted avg	0.83	0.79	0.76	123

SVM Accuracy: 0.7886178861788617

	precision	recall	f1-score	support
0	0.95	0.42	0.58	43
1	0.76	0.99	0.86	80
accuracy			0.79	123
macro avg	0.85	0.70	0.72	123
weighted avg	0.83	0.79	0.76	123

In []: