In [6]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score

file_path = r"C:\Users\s monisi prabha\Downloads\credit_customers (DS).csv"
data = pd.read_csv(file_path)

data.drop_duplicates(inplace=True)

categorical_columns = data.select_dtypes(include=['object']).columns.tolist()
categorical_columns.remove('class')
data_encoded = pd.get_dummies(data, columns=categorical_columns, drop_first=True)

X = data_encoded.drop(columns=['class'])
y = data_encoded['class'].apply(lambda x: 1 if x == 'good' else 0)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_stat

model_lr = LogisticRegression(max_iter=1000)
model_lr.fit(X_train, y_train)
y_pred_lr = model_lr.predict(X_test)

model_knn = KNeighborsClassifier(n_neighbors=5)
model_knn.fit(X_train, y_train)
y_pred_knn = model_knn.predict(X_test)

model_svm_linear = SVC(kernel='linear')
model_svm_linear.fit(X_train, y_train)
y_pred_svm_linear = model_svm_linear.predict(X_test)

model_svm_rbf = SVC(kernel='rbf')
model_svm_rbf.fit(X_train, y_train)
y_pred_svm_rbf = model_svm_rbf.predict(X_test)

print(confusion_matrix(y_test, y_pred_lr))
print(classification_report(y_test, y_pred_lr))

print(confusion_matrix(y_test, y_pred_knn))
print(classification_report(y_test, y_pred_knn))

print(confusion_matrix(y_test, y_pred_svm_linear))
print(classification_report(y_test, y_pred_svm_linear))

print(confusion_matrix(y_test, y_pred_svm_rbf))
print(classification_report(y_test, y_pred_svm_rbf))

acc_lr = accuracy_score(y_test, y_pred_lr)
print(f"\nLogistic Regression Accuracy: {acc_lr:.4f}")

acc_knn = accuracy_score(y_test, y_pred_knn)
print(f"KNN Accuracy: {acc_knn:.4f}")

acc_svm_linear = accuracy_score(y_test, y_pred_svm_linear)
print(f"SVM Linear Accuracy: {acc_svm_linear:.4f}")

acc_svm_rbf = accuracy_score(y_test, y_pred_svm_rbf)
print(f"SVM RBF Accuracy: {acc_svm_rbf:.4f}")
```

```
best_model = max(acc_lr, acc_knn, acc_svm_linear, acc_svm_rbf)
print(f"\nBest Model Accuracy: {best_model:.4f}")
```

```
[[ 32  27]
 [ 16 125]]
              precision    recall  f1-score   support

           0       0.67      0.54      0.60        59
           1       0.82      0.89      0.85       141

    accuracy                           0.79       200
   macro avg       0.74      0.71      0.73       200
weighted avg       0.78      0.79      0.78       200

[[ 13  46]
 [ 19 122]]
              precision    recall  f1-score   support

           0       0.41      0.22      0.29        59
           1       0.73      0.87      0.79       141

    accuracy                           0.68       200
   macro avg       0.57      0.54      0.54       200
weighted avg       0.63      0.68      0.64       200

[[ 28  31]
 [ 17 124]]
              precision    recall  f1-score   support

           0       0.62      0.47      0.54        59
           1       0.80      0.88      0.84       141

    accuracy                           0.76       200
   macro avg       0.71      0.68      0.69       200
weighted avg       0.75      0.76      0.75       200

[[  3  56]
 [  1 140]]
              precision    recall  f1-score   support

           0       0.75      0.05      0.10        59
           1       0.71      0.99      0.83       141

    accuracy                           0.71       200
   macro avg       0.73      0.52      0.46       200
weighted avg       0.72      0.71      0.61       200


Logistic Regression Accuracy: 0.7850
KNN Accuracy: 0.6750
SVM Linear Accuracy: 0.7600
SVM RBF Accuracy: 0.7150

Best Model Accuracy: 0.7850
```

In [ ]: