```
import kagglehub

# Download latest version
path = kagglehub.dataset_download("rupakroy/credit-data")

print("Path to dataset files:", path)
```

```
Downloading from https://www.kaggle.com/api/v1/datasets/download/rupakroy/credit-data?dataset_version_number=1...
100%|██████████| 41.1k/41.1k [00:00<00:00, 24.9MB/s]Extracting files...
Path to dataset files: /root/.cache/kagglehub/datasets/rupakroy/credit-data/versions/1
```

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, roc_auc_score
from sklearn.preprocessing import StandardScaler, LabelEncoder
from imblearn.over_sampling import SMOTE

# Load dataset
# If 'credit_data.csv' is in a different directory, replace 'credit_data.csv' with the full path to the file
# For example: data = pd.read_csv('/path/to/your/file/credit_data.csv')
data = pd.read_csv('credit_data.csv')

# Preprocessing
data.dropna(inplace=True)
X = data.drop('default', axis=1)
y = LabelEncoder().fit_transform(data['default'])

# Balance dataset
smote = SMOTE()
X_res, y_res = smote.fit_resample(X, y)

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

# Scale features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train models
log_reg = LogisticRegression()
rf = RandomForestClassifier(random_state=42)

log_reg.fit(X_train, y_train)
rf.fit(X_train, y_train)

# Evaluate models
log_reg_preds = log_reg.predict(X_test)
rf_preds = rf.predict(X_test)

print("Logistic Regression:")
print(classification_report(y_test, log_reg_preds))
print("ROC-AUC:", roc_auc_score(y_test, log_reg.predict_proba(X_test)[:, 1]))

print("\nRandom Forest:")
print(classification_report(y_test, rf_preds))
print("ROC-AUC:", roc_auc_score(y_test, rf.predict_proba(X_test)[:, 1]))
```

```
Logistic Regression:
              precision    recall  f1-score   support

           0       0.96      0.93      0.95       355
           1       0.93      0.96      0.95       331

    accuracy                           0.95       686
   macro avg       0.95      0.95      0.95       686
weighted avg       0.95      0.95      0.95       686

ROC-AUC: 0.9850304242372665

Random Forest:
              precision    recall  f1-score   support

           0       0.99      0.99      0.99       355
```

```
           1       0.99      0.99      0.99       331

    accuracy                           0.99       686
   macro avg       0.99      0.99      0.99       686
weighted avg       0.99      0.99      0.99       686

ROC-AUC: 0.9997914982341176
```