Project 3: Financial Fraud Detection

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix

# Set seed for reproducibility
np.random.seed(42)

# Number of transactions
num_transactions = 10000

# Generating synthetic data
transaction_data = {
    'TransactionID': range(1, num_transactions + 1),
    'Amount': np.random.normal(loc=100, scale=50, size=num_transactions),
    'Merchant': np.random.choice(['MerchantA', 'MerchantB', 'MerchantC'], size=num_transacti
    'IsFraud': np.random.choice([0, 1], size=num_transactions, p=[0.95, 0.05]),
}

# Creating a DataFrame
df = pd.DataFrame(transaction_data)

# Split data into features (X) and target variable (y)
X = df[['Amount']]
y = df['IsFraud']

# 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)

# Train Logistic Regression model
logreg_model = LogisticRegression(random_state=42)
logreg_model.fit(X_train, y_train)

# Train Random Forest model
rf_model = RandomForestClassifier(random_state=42)
rf_model.fit(X_train, y_train)

# Evaluate models
logreg_predictions = logreg_model.predict(X_test)
rf_predictions = rf_model.predict(X_test)

print("Logistic Regression Accuracy:", accuracy_score(y_test, logreg_predictions))
```

```
print("Random Forest Accuracy:", accuracy_score(y_test, rf_predictions))

print("\nLogistic Regression Classification Report:")
print(classification_report(y_test, logreg_predictions))

print("\nRandom Forest Classification Report:")
print(classification_report(y_test, rf_predictions))

# Save the data to a CSV file
df.to_csv('financial_transaction_data.csv', index=False)
```

```
Logistic Regression Accuracy: 0.942
Random Forest Accuracy: 0.9025

Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       0.94      1.00      0.97      1884
           1       0.00      0.00      0.00       116

    accuracy                           0.94      2000
   macro avg       0.47      0.50      0.49      2000
weighted avg       0.89      0.94      0.91      2000


Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.96      0.95      1884
           1       0.06      0.04      0.05       116

    accuracy                           0.90      2000
   macro avg       0.50      0.50      0.50      2000
weighted avg       0.89      0.90      0.90      2000


C:\Users\jilal\Anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: Undefine
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\jilal\Anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: Undefine
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\jilal\Anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: Undefine
  _warn_prf(average, modifier, msg_start, len(result))
```

This script generates synthetic data for financial transactions with features like transaction amount, merchant, and a binary indicator for fraud. The models are trained using Logistic Regression and Random Forest, and their performance is evaluated on a test set. Adjust the parameters as needed for your specific use

case.

Interpretation:

The provided code generates synthetic financial transaction data, creates a DataFrame, and trains two machine learning models (Logistic Regression and Random Forest) to predict whether a transaction is fraudulent (IsFraud = 1) based on the transaction amount. Here's the interpretation of the outputs:

1. **Accuracy Scores:**
   - Logistic Regression Accuracy: The accuracy of the Logistic Regression model on the test set.
   - Random Forest Accuracy: The accuracy of the Random Forest model on the test set.

2. **Classification Reports:**
   - Logistic Regression Classification Report: Precision, recall, F1-score, and support for both classes (fraud and non-fraud) using the Logistic Regression model.
   - Random Forest Classification Report: Precision, recall, F1-score, and support for both classes using the Random Forest model. These outputs provide a comprehensive evaluation of the models' performance, including their ability to correctly classify fraudulent and non-fraudulent transactions. The classification report metrics offer insights into precision (accuracy of positive predictions), recall (sensitivity or true positive rate), and F1-score (harmonic mean of precision and recall), giving a more nuanced view of model performance beyond simple accuracy.

Conclusion:

The Logistic Regression model achieves an accuracy of 94.2%, accurately classifying non-fraudulent transactions (class 0) but showing limitations in identifying fraudulent transactions (class 1), with low precision, recall, and F1-score for class 1. On the other hand, the Random Forest model achieves a slightly lower accuracy of 90.25%. It exhibits better performance in identifying non-fraudulent transactions but struggles with classifying fraudulent transactions, resulting in low precision, recall, and F1-score for class 1. In summary, both models excel in identifying non-fraudulent transactions but face challenges in accurately detecting fraudulent transactions, suggesting the need for further model refinement or alternative approaches for handling imbalanced datasets.