In [18]:
```
import pandas as pd
import numpy as np
```

In [19]:
```
df = pd.read_csv('payment_fraud.csv')
```

In [20]:
```
df
```

Out[20]:

	accountAgeDays	numItems	localTime	paymentMethod	paymentMethodAgeDays	label
0	29	1	4.745402	paypal	28.204861	0
1	725	1	4.742303	storecredit	0.000000	0
2	845	1	4.921318	creditcard	0.000000	0
3	503	1	4.886641	creditcard	0.000000	0
4	2000	1	5.040929	creditcard	0.000000	0
...	...	...	...	...	...	...
39216	986	1	4.836982	creditcard	0.000000	0
39217	1647	1	4.876771	creditcard	377.930556	0
39218	1591	1	4.742303	creditcard	0.000000	0
39219	237	1	4.921318	creditcard	236.082639	0
39220	272	1	5.040929	paypal	0.000694	0

39221 rows × 6 columns

In [21]:
```
df.isnull()
```

Out[21]:

	accountAgeDays	numItems	localTime	paymentMethod	paymentMethodAgeDays	label
0	False	False	False	False	False	False
1	False	False	False	False	False	False
2	False	False	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
...	...	...	...	...	...	...
39216	False	False	False	False	False	False
39217	False	False	False	False	False	False
39218	False	False	False	False	False	False
39219	False	False	False	False	False	False
39220	False	False	False	False	False	False

39221 rows × 6 columns

In [22]:
```
df1 = (df['label'] == 1)
```

In [23]:
```
df1
```

Out[23]:
```
0        False
1        False
2        False
3        False
4        False
         ...
39216    False
39217    False
39218    False
39219    False
39220    False
Name: label, Length: 39221, dtype: bool
```

In [24]:
```
df1.unique()
```

Out[24]:
```
array([False,  True])
```

In [25]:
```
df1.count()
```

Out[25]:
```
39221
```

In [26]:
```
df1.value_counts()
```

Out[26]:
```
False    38661
True       560
Name: label, dtype: int64
```

In [27]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
```

In [28]:
```
le = LabelEncoder()
df['paymentMethod'] = le.fit_transform(df['paymentMethod'])
```

In [29]:
```
X = df.drop('label', axis=1)
y = df['label']
```

In [30]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=
```

In [31]:
```
classifiers = {
    'Logistic Regression': LogisticRegression(),
    'Random Forest': RandomForestClassifier(),
    'SVM': SVC()
}
```

In [32]:
```
results = {}
for name, classifier in classifiers.items():
    classifier.fit(X_train, y_train)
    y_pred = classifier.predict(X_test)
```

```
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)

    results[name] = {
        'Accuracy': accuracy,
        'Precision': precision,
        'Recall': recall,
        'F1-score': f1
    }
```

```
D:\Anaconda\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetri
cWarning: Precision is ill-defined and being set to 0.0 due to no predicted samples.
Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [33]:
```
print("Classification Results:")
for name, metrics in results.items():
    print(f"{name}:")
    print(f"  Accuracy: {metrics['Accuracy']:.4f}")
    print(f"  Precision: {metrics['Precision']:.4f}")
    print(f"  Recall: {metrics['Recall']:.4f}")
    print(f"  F1-score: {metrics['F1-score']:.4f}")
    print("="*30)
```

```
Classification Results:
Logistic Regression:
  Accuracy: 1.0000
  Precision: 1.0000
  Recall: 1.0000
  F1-score: 1.0000
==============================
Random Forest:
  Accuracy: 1.0000
  Precision: 1.0000
  Recall: 1.0000
  F1-score: 1.0000
==============================
SVM:
  Accuracy: 0.9850
  Precision: 0.0000
  Recall: 0.0000
  F1-score: 0.0000
==============================
```

In [ ]: