In [1]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
```

In [2]:
```
data = pd.read_csv("payment_fraud.csv")
data
```

Out[2]:

	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 [3]:
```
data.label.value_counts(normalize=True)
```

Out[3]:
```
0    0.985722
1    0.014278
Name: label, dtype: float64
```

In [4]:
```
data.paymentMethod.value_counts(normalize=True)
```

Out[4]:
```
creditcard     0.714005
paypal         0.237194
storecredit    0.048800
Name: paymentMethod, dtype: float64
```

In [5]:
```
data['paymentMethod'].replace(['creditcard','paypal','storecredit'],[2,1,0],inplace=True)
data
```

Out[5]:

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

39221 rows × 6 columns

In [6]:
```
X = data.drop('label', axis=1)  # assuming 'target' is the name of your dependent variable
y = data['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

In [7]:
```
models = {
    'Logistic Regression': LogisticRegression(),
    'Decision Tree': DecisionTreeClassifier(),
    'SVM': SVC()
}

for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    accuracy = accuracy_score(y_test, y_pred)
    report = classification_report(y_test, y_pred)

    print(confusion_matrix(y_test, y_pred))
    print(classification_report(y_test, y_pred))
    print(f'Model: {name}')
    print(f'Accuracy: {accuracy}')
    print(f'Classification Report:\n{report}\n')
```

```
[[11595     0]
 [    0   172]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     11595
           1       1.00      1.00      1.00       172

    accuracy                           1.00     11767
   macro avg       1.00      1.00      1.00     11767
weighted avg       1.00      1.00      1.00     11767

Model: Logistic Regression
Accuracy: 1.0
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     11595
           1       1.00      1.00      1.00       172

    accuracy                           1.00     11767
   macro avg       1.00      1.00      1.00     11767
weighted avg       1.00      1.00      1.00     11767


[[11595     0]
 [    0   172]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     11595
           1       1.00      1.00      1.00       172

    accuracy                           1.00     11767
   macro avg       1.00      1.00      1.00     11767
weighted avg       1.00      1.00      1.00     11767

Model: Decision Tree
Accuracy: 1.0
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     11595
           1       1.00      1.00      1.00       172

    accuracy                           1.00     11767
   macro avg       1.00      1.00      1.00     11767
weighted avg       1.00      1.00      1.00     11767


[[11595     0]
 [  172     0]]
              precision    recall  f1-score   support

           0       0.99      1.00      0.99     11595
           1       0.00      0.00      0.00       172

    accuracy                           0.99     11767
   macro avg       0.49      0.50      0.50     11767
weighted avg       0.97      0.99      0.98     11767

Model: SVM
Accuracy: 0.9853828503441829
Classification Report:
              precision    recall  f1-score   support

           0       0.99      1.00      0.99     11595
           1       0.00      0.00      0.00       172

    accuracy                           0.99     11767
   macro avg       0.49      0.50      0.50     11767
weighted avg       0.97      0.99      0.98     11767

```

```
C:\Users\supra\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\supra\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\supra\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\supra\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\supra\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\supra\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1344: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [ ]: