In [3]:
```
import pandas as pd
fraud = pd.read_csv("payment_fraud.csv")
fraud
```

Out[3]:

	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 [5]:
```
print(fraud.columns)
```

```
Index(['accountAgeDays', 'numItems', 'localTime', 'paymentMethod',
       'paymentMethodAgeDays', 'label'],
      dtype='object')
```

In [6]:
```
print(fraud.head())
```

```
   accountAgeDays  numItems  localTime paymentMethod  paymentMethodAgeDays  \
0              29         1   4.745402        paypal             28.204861
1             725         1   4.742303   storecredit              0.000000
2             845         1   4.921318    creditcard              0.000000
3             503         1   4.886641    creditcard              0.000000
4            2000         1   5.040929    creditcard              0.000000

   label
0      0
1      0
2      0
3      0
4      0
```

In [32]:
```
print(fraud.groupby('label').size())
```

```
label
0    38661
1      560
dtype: int64
```

In [20]:
```
feature_names=['accountAgeDays','numItems','localTime','paymentMethodAgeDays']
X = fraud[feature_names]
y = fraud.label
```

In [21]:
```
X
```

Out[21]:

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

39221 rows × 4 columns

In [22]:
```
y
```

Out[22]:
```
0        0
1        0
2        0
3        0
4        0
        ..
39216    0
39217    0
39218    0
39219    0
39220    0
Name: label, Length: 39221, dtype: int64
```

In [23]:
```
import warnings;
warnings.simplefilter('ignore')
```

In [24]:
```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
```

In [25]:
```
models = []
models.append(('KNN', KNeighborsClassifier()))
models.append(('DT', DecisionTreeClassifier()))
models.append(('GNB', GaussianNB()))
```

In [26]:
```
models
```

Out[26]:
```
[('KNN', KNeighborsClassifier()),
 ('DT', DecisionTreeClassifier()),
 ('GNB', GaussianNB())]
```

In [27]:
```
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

X_train, X_test, y_train, y_test = train_test_split(X, y, stratify = fraud.label)
```

In [28]:
```
names = []
scores = []
```

```
for name, model in models:
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    scores.append(accuracy_score(y_test, y_pred))
    names.append(name)
tr_split = pd.DataFrame({'Name': names, 'Score': scores})
print(tr_split)
```

```
  Name     Score
0  KNN  0.999898
1   DT  1.000000
2  GNB  1.000000
```

In [29]:
```
from sklearn.metrics import classification_report, confusion_matrix
print(confusion_matrix(y_test, y_pred))
```

```
[[9666    0]
 [   0  140]]
```

In [30]:
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      9666
           1       1.00      1.00      1.00       140

    accuracy                           1.00      9806
   macro avg       1.00      1.00      1.00      9806
weighted avg       1.00      1.00      1.00      9806
```

In [ ]:
```

```