In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
```

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

In [3]:
```
data.head()
```

Out[3]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141

5 rows × 31 columns

In [4]:
```
fraud = data.loc[data['Class']==1]
normal = data.loc[data['Class']==0]
```

In [5]:
```
len(fraud)
```

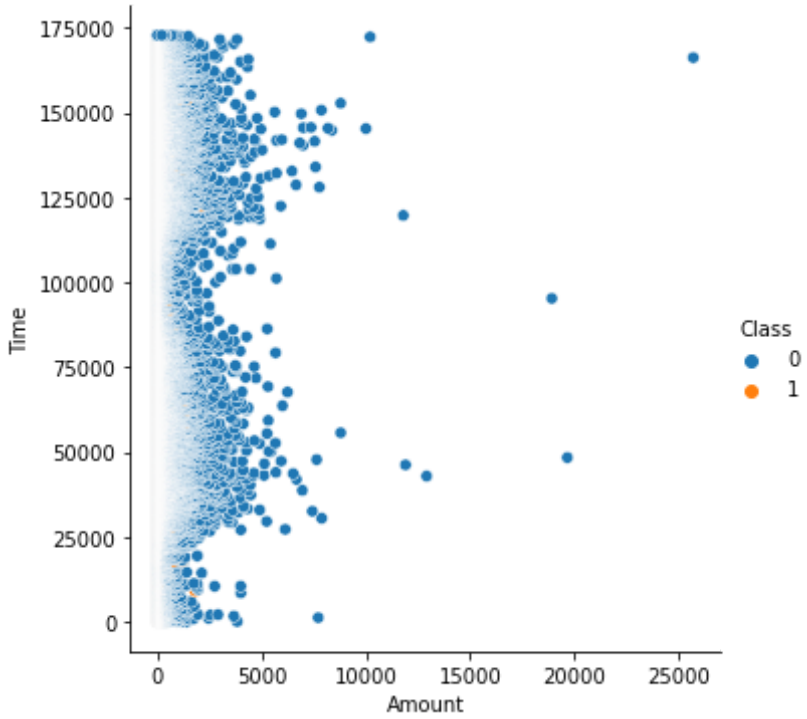
Out[5]:
```
492
```

In [6]:
```
len(normal)
```

Out[6]:
```
284315
```

In [7]:
```
sns.relplot(x="Amount", y="Time",hue="Class", data=data)
```

Out[7]:
```
<seaborn.axisgrid.FacetGrid at 0x80fedf0>
```



In [8]:
```
from sklearn import linear_model
from sklearn.model_selection import train_test_split
```

In [11]:
```
X = data.iloc[:,:-1]
y = data['Class']
```

In [12]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.35)
```

In [15]:
```
clf = linear_model.LogisticRegression(C=1e5)
```

In [16]:
```
clf.fit(X_train, y_train)
```

```
C:\Users\win7\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:762: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.
```

```
Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

Out[16]:
```
LogisticRegression(C=100000.0)
```

In [18]:
```
y_pred = np.array(clf.predict(X_test))
y = np.array(y_test)
```

In [19]:
```
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
```

In [20]:
```
print(confusion_matrix(y_test, y_pred))
```

```
[[99473    43]
 [   40   127]]
```

In [21]:
```
print(accuracy_score(y_test,y_pred))
```

```
0.9991673605328892
```

In [22]:
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     99516
           1       0.75      0.76      0.75       167

    accuracy                           1.00     99683
   macro avg       0.87      0.88      0.88     99683
weighted avg       1.00      1.00      1.00     99683
```

In [ ]:
```

```