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

In [2]:
```
data=pd.read_csv("Documents/MLProject/Project3/creditcard.csv")
data.head()
```

Out[2]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798

5 rows × 31 columns

In [3]:
```
fraud=data.loc[data['Class']==1]
normal=data.loc[data['Class']==0]
```

In [4]:
```
len(fraud)
```

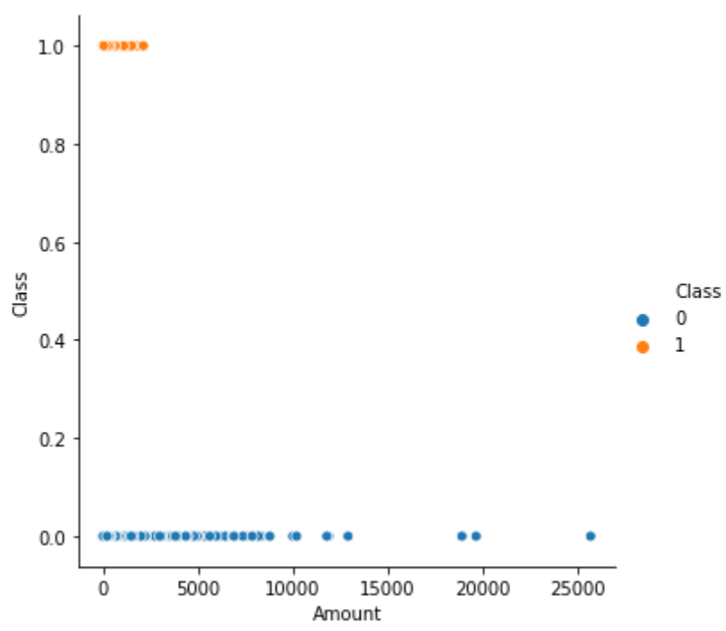
Out[4]:
```
492
```

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

Out[5]:
```
284315
```

In [6]:
```
sns.relplot(x='Amount', y='Class', hue='Class', data=data)
```

Out[6]:
```
<seaborn.axisgrid.FacetGrid at 0x1a221bc510>
```



In [7]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
```

In [8]:
```
x=data.drop(['Class'], axis='columns')
y=data.Class
```

In [9]:
```
x_train, x_test, y_train, y_test=train_test_split(x,y,test_size=0.25)
```

In [10]:
```
model1=LogisticRegression()
model1.fit(x_train,y_train)
```

```
/opt/anaconda3/lib/python3.7/site-packages/sklearn/linear_model/_logistic.py:940: 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
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
```

Out[10]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

In [11]:
```
model2=SVC()
model2.fit(x_train,y_train)
```

Out[11]:
```
SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

In [12]:
```
y_predict=model1.predict(x_test)
y=np.array(y_test)
```

In [13]:
```
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
```

In [14]:
```
print(confusion_matrix(y_test, y_predict))
```

```
[[71030    47]
 [   31    94]]
```

In [15]:
```
print(accuracy_score(y_test, y_predict))
```

```
0.9989045251537878
```

In [16]:
```
model1.score(x_test, y_test)
```

Out[16]:
```
0.9989045251537878
```

In [18]:
```
model2.score(x_test, y_test)
```

Out[18]:
```
0.9982444313361984
```

In [17]:
```
print(classification_report(y_test,y_predict))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     71077
           1       0.67      0.75      0.71       125

    accuracy                           1.00     71202
   macro avg       0.83      0.88      0.85     71202
weighted avg       1.00      1.00      1.00     71202
```

In [ ]:
```

```