lab4

December 3, 2019

[2]:
```
from sklearn import datasets
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np
import seaborn as sns
```

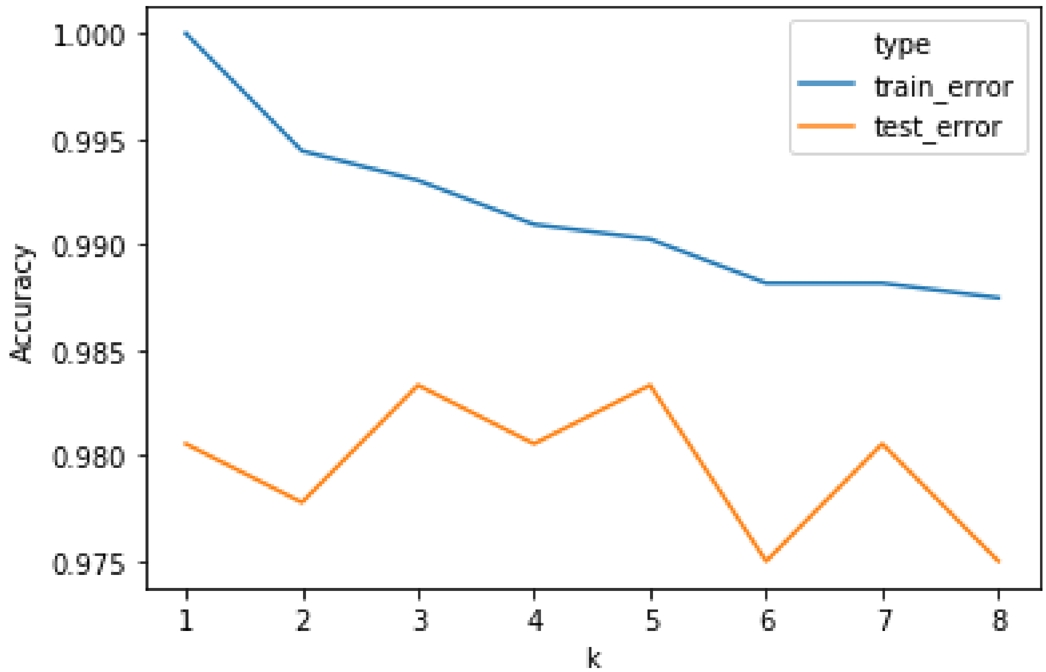
[3]:
```
digits = datasets.load_digits()
data = digits['data']
target = digits['target']
#len(data)
```

[4]:
```
data_train,data_test,target_train,target_test  =␣
 ↪train_test_split(data,target,test_size=0.2)
```

[5]:
```
from sklearn.neighbors import KNeighborsClassifier
import seaborn as sns
error_pd = pd.DataFrame(columns = ['k','Accuracy','type'])

for k in range(1,9):
    error_1={}
    error_2={}
    neigh = KNeighborsClassifier(n_neighbors=k)
    neigh.fit(data_train,target_train)
    train_error = neigh.score(data_train,target_train)
    #print(train_error)
    test_error = neigh.score(data_test,target_test)
    if(k==3):
        knn_score = test_error
    error_1['k']=k
    error_1['Accuracy']=train_error
    error_1['type'] = 'train_error'
    #print(error_1)
    error_2['k']=k
    error_2['Accuracy']=test_error
    error_2['type'] = 'test_error'
    #print(error_1)
    #error['test_error']=test_error
    error_pd = error_pd.append(error_1,ignore_index=True)
```

```
    error_pd = error_pd.append(error_2,ignore_index=True)
#error_pd
sns.lineplot(x="k",y="Accuracy",hue="type",data=error_pd)
```

[5]: <matplotlib.axes._subplots.AxesSubplot at 0x20047438470>



[6]:
```
from sklearn.svm import SVC
from sklearn.metrics import classification_report
```

[7]:
```
rbf_SVM = SVC(kernel="rbf",gamma="auto")
linear_SVM = SVC(kernel="linear",gamma="auto")
sigmoid_SVM = SVC(kernel="sigmoid",gamma="auto")
```

[8]:
```
rbf_SVM.fit(data_train,target_train)
linear_SVM.fit(data_train,target_train)
sigmoid_SVM.fit(data_train,target_train)
```

[8]:
```
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto', kernel='sigmoid',
    max_iter=-1, probability=False, random_state=None, shrinking=True,
    tol=0.001, verbose=False)
```

[9]:
```
y_1 = rbf_SVM.predict(data_test)
print(classification_report(target_test,y_1))

y_2 = linear_SVM.predict(data_test)
print(classification_report(target_test,y_2))
svm_score = linear_SVM.score(data_test,target_test)
```

```python
y_3 = sigmoid_SVM.predict(data_test)
print(classification_report(target_test,y_3))
```

```
              precision    recall  f1-score   support

           0       1.00      0.42      0.60        40
           1       1.00      0.31      0.47        39
           2       1.00      0.07      0.13        42
           3       1.00      0.36      0.53        39
           4       1.00      0.47      0.64        34
           5       0.12      1.00      0.22        30
           6       1.00      0.56      0.72        32
           7       1.00      0.41      0.58        39
           8       1.00      0.11      0.21        35
           9       1.00      0.60      0.75        30

    accuracy                           0.41       360
   macro avg       0.91      0.43      0.48       360
weighted avg       0.93      0.41      0.48       360

              precision    recall  f1-score   support

           0       1.00      1.00      1.00        40
           1       0.97      1.00      0.99        39
           2       1.00      1.00      1.00        42
           3       0.93      0.97      0.95        39
           4       0.97      1.00      0.99        34
           5       1.00      0.97      0.98        30
           6       1.00      1.00      1.00        32
           7       1.00      1.00      1.00        39
           8       1.00      0.86      0.92        35
           9       0.91      0.97      0.94        30

    accuracy                           0.98       360
   macro avg       0.98      0.98      0.98       360
weighted avg       0.98      0.98      0.98       360

              precision    recall  f1-score   support

           0       0.00      0.00      0.00        40
           1       0.00      0.00      0.00        39
           2       0.00      0.00      0.00        42
           3       0.00      0.00      0.00        39
           4       0.00      0.00      0.00        34
           5       0.08      1.00      0.15        30
           6       0.00      0.00      0.00        32
           7       0.00      0.00      0.00        39
```

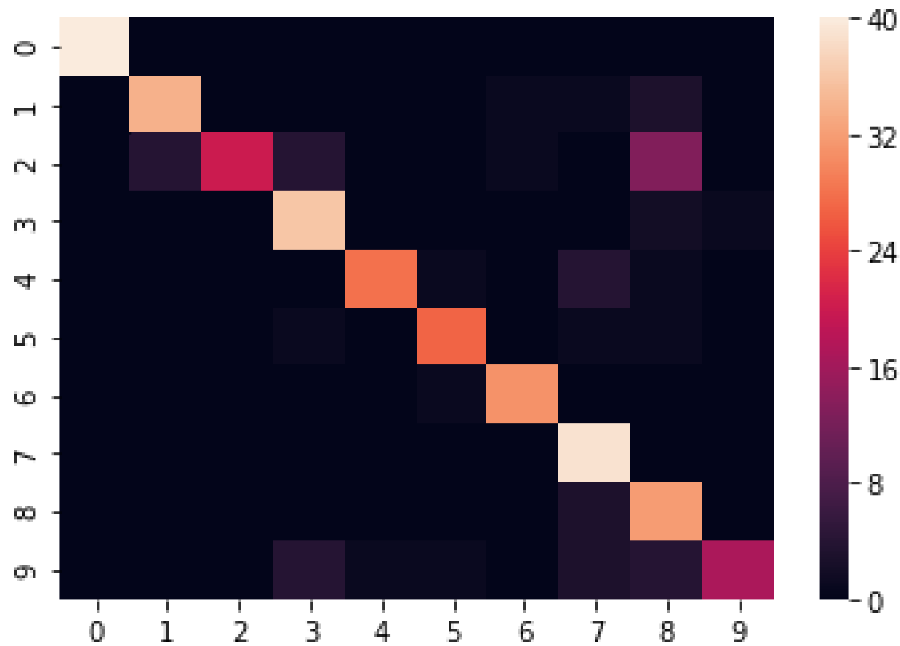
```
           8       0.00      0.00      0.00        35
           9       0.00      0.00      0.00        30

    accuracy                           0.08       360
   macro avg       0.01      0.10      0.02       360
weighted avg       0.01      0.08      0.01       360
```

```
E:\software\anaconda\lib\site-packages\sklearn\metrics\classification.py:1437:
UndefinedMetricWarning: Precision and F-score are ill-defined and being set to
0.0 in labels with no predicted samples.
  'precision', 'predicted', average, warn_for)
```

better to use linear kernel

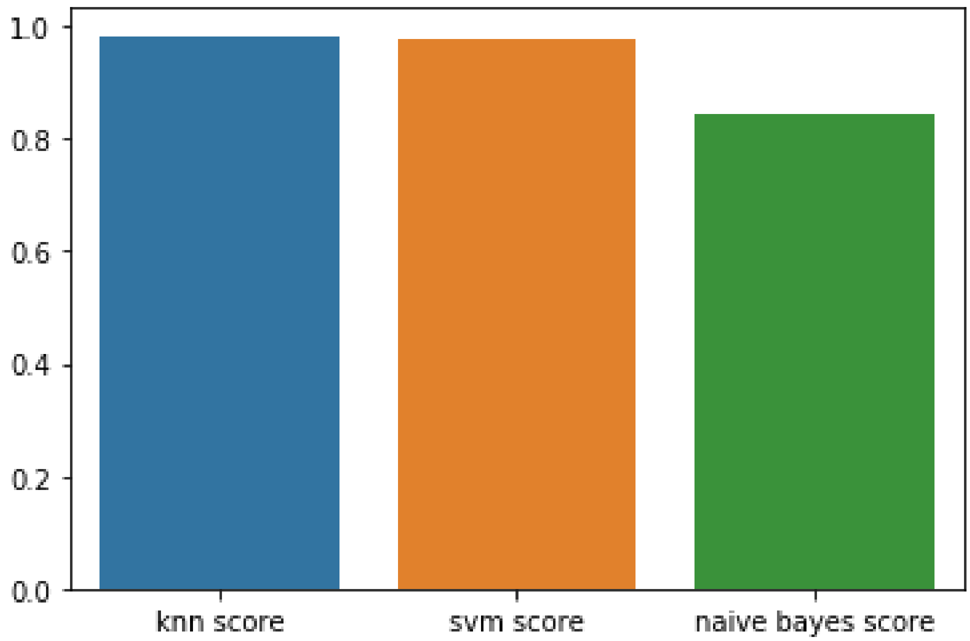
[10]:
```python
from sklearn.metrics import confusion_matrix
from sklearn.naive_bayes import GaussianNB
```

[11]:
```python
nb = GaussianNB()
nb.fit(data_train,target_train)
y = nb.predict(data_test)
confusion_data = confusion_matrix(target_test,y)
sns.heatmap(confusion_data)
nb_score = nb.score(data_test,target_test)
```



[12]:
```python
x=["knn score","svm score","naive bayes score"]
y = [knn_score,svm_score,nb_score]
sns.barplot(x,y)
```

[12]: <matplotlib.axes._subplots.AxesSubplot at 0x20048feac88>



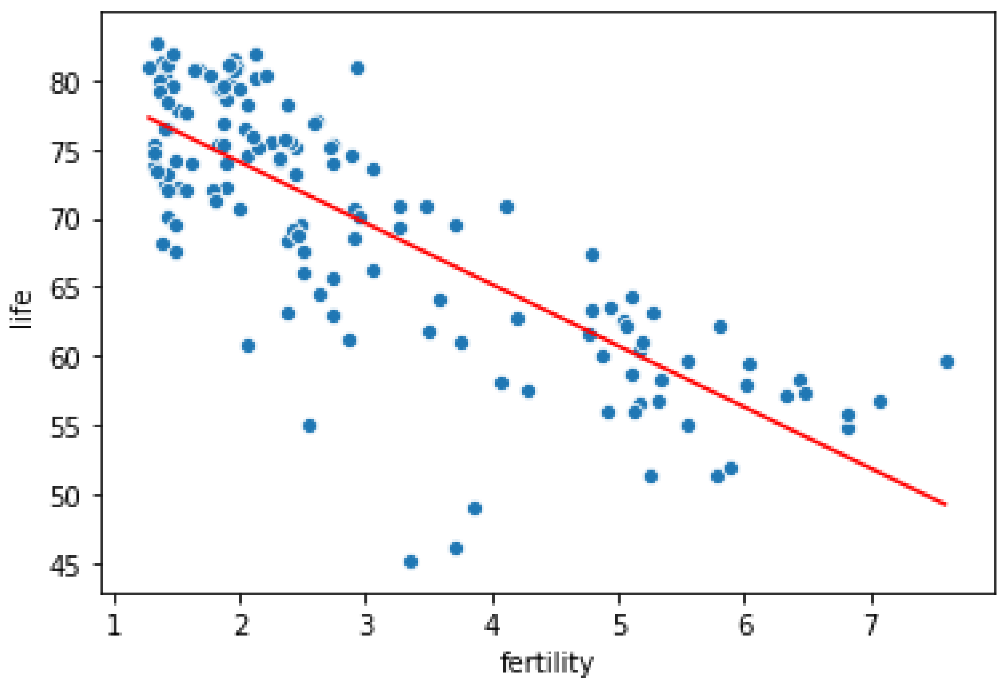
1. Linear Regression

[13]:
```
from sklearn.linear_model import LinearRegression
gap_pd = pd.read_csv("./gapminder.csv")
gap_pd.head(3)
gap_pd['Region'].unique()
```

[13]:
```
array(['Middle East & North Africa', 'Sub-Saharan Africa', 'America',
       'Europe & Central Asia', 'East Asia & Pacific', 'South Asia'],
      dtype=object)
```

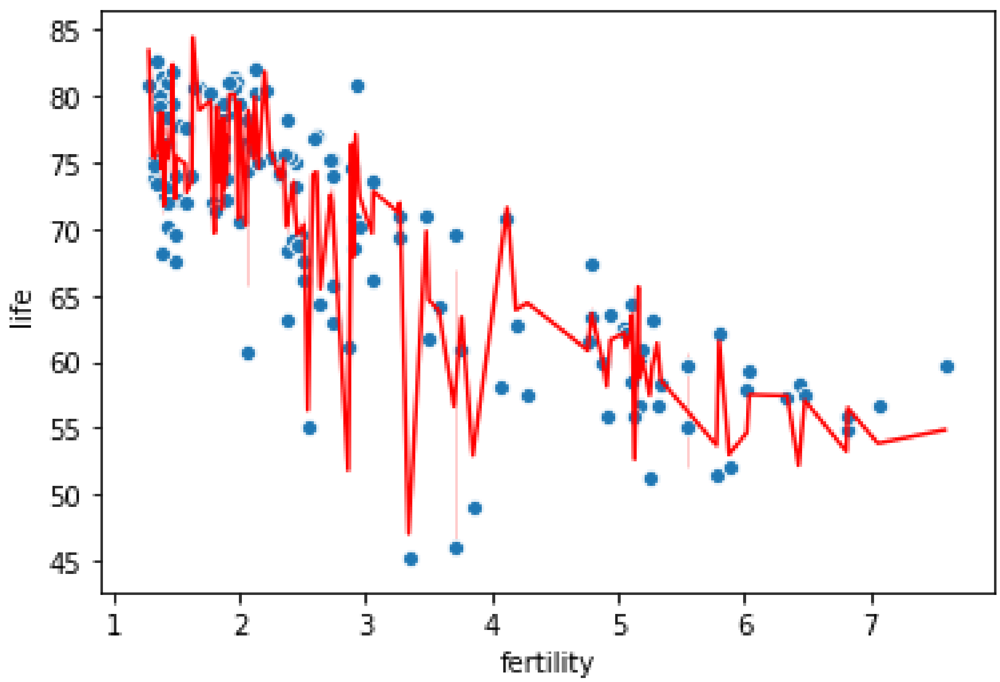
[14]:
```
lr = LinearRegression()
x = np.reshape(gap_pd['fertility'].values,(-1,1))
y = gap_pd['life']

lr.fit(x,y)
lr.coef_
lr.intercept_
y_pred = lr.predict(x)
sns.lineplot(gap_pd['fertility'],y_pred,color='red')
sns.scatterplot(gap_pd['fertility'],y)
```

[14]: <matplotlib.axes._subplots.AxesSubplot at 0x200490755f8>



[15]:
```
lr = LinearRegression()

x_arr = []
index = 0
for index,row in gap_pd.iterrows():
    row_1 = []
    for name in gap_pd.columns:
        #print(name)
        if name == 'life' or name == 'Region':
            continue
        row_1.append(row[name])
        #x_arr.append(np.reshape(gap_pd[name].values,(-1,1)))
        #x_arr = np.insert(x_arr,index,values=np.reshape(gap_pd[name].
 ↪values,(-1,1)),axis=1)
        # print(len(x_arr))
    x_arr.append(row_1)
x_arr = np.array(x_arr)
y = gap_pd['life']
#x_arr.shape
lr.fit(x_arr,y)
#lr.coef_
#lr.intercept_
y_pred = lr.predict(x_arr)
sns.lineplot(gap_pd['fertility'],y_pred,color='red')
sns.scatterplot(gap_pd['fertility'],y)
```

[15]: <matplotlib.axes._subplots.AxesSubplot at 0x200490d57b8>



[ ]:

[18]:
```python
credit_pd = pd.read_csv('./creditcard.csv')
credit_pd.head(3)
for idx,row in credit_pd.iterrows():
    if(row['Class']==0):
        row['type'] = 'fraud'
    else:
        row['type'] = 'nonfraud'
credit_pd
```

[18]:

	Time	V1	V2	V3	V4	V5	\
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	
5	2.0	-0.425966	0.960523	1.141109	-0.168252	0.420987	
6	4.0	1.229658	0.141004	0.045371	1.202613	0.191881	
7	7.0	-0.644269	1.417964	1.074380	-0.492199	0.948934	
8	7.0	-0.894286	0.286157	-0.113192	-0.271526	2.669599	
9	9.0	-0.338262	1.119593	1.044367	-0.222187	0.499361	
10	10.0	1.449044	-1.176339	0.913860	-1.375667	-1.971383	
11	10.0	0.384978	0.616109	-0.874300	-0.094019	2.924584	

12	10.0	1.249999	-1.221637	0.383930	-1.234899	-1.485419
13	11.0	1.069374	0.287722	0.828613	2.712520	-0.178398
14	12.0	-2.791855	-0.327771	1.641750	1.767473	-0.136588
15	12.0	-0.752417	0.345485	2.057323	-1.468643	-1.158394
16	12.0	1.103215	-0.040296	1.267332	1.289091	-0.735997
17	13.0	-0.436905	0.918966	0.924591	-0.727219	0.915679
18	14.0	-5.401258	-5.450148	1.186305	1.736239	3.049106
19	15.0	1.492936	-1.029346	0.454795	-1.438026	-1.555434
20	16.0	0.694885	-1.361819	1.029221	0.834159	-1.191209
21	17.0	0.962496	0.328461	-0.171479	2.109204	1.129566
22	18.0	1.166616	0.502120	-0.067300	2.261569	0.428804
23	18.0	0.247491	0.277666	1.185471	-0.092603	-1.314394
24	22.0	-1.946525	-0.044901	-0.405570	-1.013057	2.941968
25	22.0	-2.074295	-0.121482	1.322021	0.410008	0.295198
26	23.0	1.173285	0.353498	0.283905	1.133563	-0.172577
27	23.0	1.322707	-0.174041	0.434555	0.576038	-0.836758
28	23.0	-0.414289	0.905437	1.727453	1.473471	0.007443
29	23.0	1.059387	-0.175319	1.266130	1.186110	-0.786002
...	...	...	...	...	...	...
284777	172764.0	2.079137	-0.028723	-1.343392	0.358000	-0.045791
284778	172764.0	-0.764523	0.588379	-0.907599	-0.418847	0.901528
284779	172766.0	1.975178	-0.616244	-2.628295	-0.406246	2.327804
284780	172766.0	-1.727503	1.108356	2.219561	1.148583	-0.884199
284781	172766.0	-1.139015	-0.155510	1.894478	-1.138957	1.451777
284782	172767.0	-0.268061	2.540315	-1.400915	4.846661	0.639105
284783	172768.0	-1.796092	1.929178	-2.828417	-1.689844	2.199572
284784	172768.0	-0.669662	0.923769	-1.543167	-1.560729	2.833960
284785	172768.0	0.032887	0.545338	-1.185844	-1.729828	2.932315
284786	172768.0	-2.076175	2.142238	-2.522704	-1.888063	1.982785
284787	172769.0	-1.029719	-1.110670	-0.636179	-0.840816	2.424360
284788	172770.0	2.007418	-0.280235	-0.208113	0.335261	-0.715798
284789	172770.0	-0.446951	1.302212	-0.168583	0.981577	0.578957
284790	172771.0	-0.515513	0.971950	-1.014580	-0.677037	0.912430
284791	172774.0	-0.863506	0.874701	0.420358	-0.530365	0.356561
284792	172774.0	-0.724123	1.485216	-1.132218	-0.607190	0.709499
284793	172775.0	1.971002	-0.699067	-1.697541	-0.617643	1.718797
284794	172777.0	-1.266580	-0.400461	0.956221	-0.723919	1.531993
284795	172778.0	-12.516732	10.187818	-8.476671	-2.510473	-4.586669
284796	172780.0	1.884849	-0.143540	-0.999943	1.506772	-0.035300
284797	172782.0	-0.241923	0.712247	0.399806	-0.463406	0.244531
284798	172782.0	0.219529	0.881246	-0.635891	0.960928	-0.152971
284799	172783.0	-1.775135	-0.004235	1.189786	0.331096	1.196063
284800	172784.0	2.039560	-0.175233	-1.196825	0.234580	-0.008713
284801	172785.0	0.120316	0.931005	-0.546012	-0.745097	1.130314
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515

```
284805  172788.0  -0.240440   0.530483  0.702510  0.689799 -0.377961
284806  172792.0  -0.533413  -0.189733  0.703337 -0.506271 -0.012546

              V6        V7        V8        V9  ...       V21       V22  \
0       0.462388  0.239599  0.098698  0.363787  ... -0.018307  0.277838
1      -0.082361 -0.078803  0.085102 -0.255425  ... -0.225775 -0.638672
2       1.800499  0.791461  0.247676 -1.514654  ...  0.247998  0.771679
3       1.247203  0.237609  0.377436 -1.387024  ... -0.108300  0.005274
4       0.095921  0.592941 -0.270533  0.817739  ... -0.009431  0.798278
5      -0.029728  0.476201  0.260314 -0.568671  ... -0.208254 -0.559825
6       0.272708 -0.005159  0.081213  0.464960  ... -0.167716 -0.270710
7       0.428118  1.120631 -3.807864  0.615375  ...  1.943465 -1.015455
8       3.721818  0.370145  0.851084 -0.392048  ... -0.073425 -0.268092
9      -0.246761  0.651583  0.069539 -0.736727  ... -0.246914 -0.633753
10     -0.629152 -1.423236  0.048456 -1.720408  ... -0.009302  0.313894
11      3.317027  0.470455  0.538247 -0.558895  ...  0.049924  0.238422
12     -0.753230 -0.689405 -0.227487 -2.094011  ... -0.231809 -0.483285
13      0.337544 -0.096717  0.115982 -0.221083  ... -0.036876  0.074412
14      0.807596 -0.422911 -1.907107  0.755713  ...  1.151663  0.222182
15     -0.077850 -0.608581  0.003603 -0.436167  ...  0.499625  1.353650
16      0.288069 -0.586057  0.189380  0.782333  ... -0.024612  0.196002
17     -0.127867  0.707642  0.087962 -0.665271  ... -0.194796 -0.672638
18     -1.763406 -1.559738  0.160842  1.233090  ... -0.503600  0.984460
19     -0.720961 -1.080664 -0.053127 -1.978682  ... -0.177650 -0.175074
20      1.309109 -0.878586  0.445290 -0.446196  ... -0.295583 -0.571955
21      1.696038  0.107712  0.521502 -1.191311  ...  0.143997  0.402492
22      0.089474  0.241147  0.138082 -0.989162  ...  0.018702 -0.061972
23     -0.150116 -0.946365 -1.617935  1.544071  ...  1.650180  0.200454
24      2.955053 -0.063063  0.855546  0.049967  ... -0.579526 -0.799229
25     -0.959537  0.543985 -0.104627  0.475664  ... -0.403639 -0.227404
26     -0.916054  0.369025 -0.327260 -0.246651  ...  0.067003  0.227812
27     -0.831083 -0.264905 -0.220982 -1.071425  ... -0.284376 -0.323357
28     -0.200331  0.740228 -0.029247 -0.593392  ...  0.077237  0.457331
29      0.578435 -0.767084  0.401046  0.699500  ...  0.013676  0.213734
...          ...       ...       ...       ...  ...       ...       ...
284777 -1.345452  0.227476 -0.378355  0.665911  ...  0.235758  0.829758
284778 -0.760802  0.758545  0.414698 -0.730854  ...  0.003530 -0.431876
284779  3.664740 -0.533297  0.842937  1.128798  ...  0.086043  0.543613
284780  0.793083 -0.527298  0.866429  0.853819  ... -0.094708  0.236818
284781  0.093598  0.191353  0.092211 -0.062621  ... -0.191027 -0.631658
284782  0.186479 -0.045911  0.936448 -2.419986  ... -0.263889 -0.857904
284783  3.123732 -0.270714  1.657495  0.465804  ...  0.271170  1.145750
284784  3.240843  0.181576  1.282746 -0.893890  ...  0.183856  0.202670
284785  3.401529  0.337434  0.925377 -0.165663  ... -0.266113 -0.716336
284786  3.732950 -1.217430 -0.536644  0.272867  ...  2.016666 -1.588269
284787 -2.956733  0.283610 -0.332656 -0.247488  ...  0.353722  0.488487
284788 -0.751373 -0.458972 -0.140140  0.959971  ... -0.208260 -0.430347
```

```
284789 -0.605641  1.253430 -1.042610 -0.417116  ...  0.851800  0.305268
284790 -0.316187  0.396137  0.532364 -0.224606  ... -0.280302 -0.849919
284791 -1.046238  0.757051  0.230473 -0.506856  ... -0.108846 -0.480820
284792 -0.482638  0.548393  0.343003 -0.226323  ...  0.414621  1.307511
284793  3.911336 -1.259306  1.056209  1.315006  ...  0.188758  0.694418
284794 -1.788600  0.314741  0.004704  0.013857  ... -0.157831 -0.883365
284795 -1.394465 -3.632516  5.498583  4.893089  ... -0.944759 -1.565026
284796 -0.613638  0.190241 -0.249058  0.666458  ...  0.144008  0.634646
284797 -1.343668  0.929369 -0.206210  0.106234  ... -0.228876 -0.514376
284798 -1.014307  0.427126  0.121340 -0.285670  ...  0.099936  0.337120
284799  5.519980 -1.518185  2.080825  1.159498  ...  0.103302  0.654850
284800 -0.726571  0.017050 -0.118228  0.435402  ... -0.268048 -0.717211
284801 -0.235973  0.812722  0.115093 -0.204064  ... -0.314205 -0.808520
284802 -2.606837 -4.918215  7.305334  1.914428  ...  0.213454  0.111864
284803  1.058415  0.024330  0.294869  0.584800  ...  0.214205  0.924384
284804  3.031260 -0.296827  0.708417  0.432454  ...  0.232045  0.578229
284805  0.623708 -0.686180  0.679145  0.392087  ...  0.265245  0.800049
284806 -0.649617  1.577006 -0.414650  0.486180  ...  0.261057  0.643078

             V23       V24       V25       V26       V27       V28  Amount  \
0      -0.110474  0.066928  0.128539 -0.189115  0.133558 -0.021053  149.62
1       0.101288 -0.339846  0.167170  0.125895 -0.008983  0.014724    2.69
2       0.909412 -0.689281 -0.327642 -0.139097 -0.055353 -0.059752  378.66
3      -0.190321 -1.175575  0.647376 -0.221929  0.062723  0.061458  123.50
4      -0.137458  0.141267 -0.206010  0.502292  0.219422  0.215153   69.99
5      -0.026398 -0.371427 -0.232794  0.105915  0.253844  0.081080    3.67
6      -0.154104 -0.780055  0.750137 -0.257237  0.034507  0.005168    4.99
7       0.057504 -0.649709 -0.415267 -0.051634 -1.206921 -1.085339   40.80
8      -0.204233  1.011592  0.373205 -0.384157  0.011747  0.142404   93.20
9      -0.120794 -0.385050 -0.069733  0.094199  0.246219  0.083076    3.68
10      0.027740  0.500512  0.251367 -0.129478  0.042850  0.016253    7.80
11      0.009130  0.996710 -0.767315 -0.492208  0.042472 -0.054337    9.99
12      0.084668  0.392831  0.161135 -0.354990  0.026416  0.042422  121.50
13     -0.071407  0.104744  0.548265  0.104094  0.021491  0.021293   27.50
14      1.020586  0.028317 -0.232746 -0.235557 -0.164778 -0.030154   58.80
15     -0.256573 -0.065084 -0.039124 -0.087086 -0.180998  0.129394   15.99
16      0.013802  0.103758  0.364298 -0.382261  0.092809  0.037051   12.99
17     -0.156858 -0.888386 -0.342413 -0.049027  0.079692  0.131024    0.89
18      2.458589  0.042119 -0.481631 -0.621272  0.392053  0.949594   46.80
19      0.040002  0.295814  0.332931 -0.220385  0.022298  0.007602    5.00
20     -0.050881 -0.304215  0.072001 -0.422234  0.086553  0.063499  231.71
21     -0.048508 -1.371866  0.390814  0.199964  0.016371 -0.014605   34.09
22     -0.103855 -0.370415  0.603200  0.108556 -0.040521 -0.011418    2.28
23     -0.185353  0.423073  0.820591 -0.227632  0.336634  0.250475   22.75
24      0.870300  0.983421  0.321201  0.149650  0.707519  0.014600    0.89
25      0.742435  0.398535  0.249212  0.274404  0.359969  0.243232   26.43
26     -0.150487  0.435045  0.724825 -0.337082  0.016368  0.030041   41.88
```

```
27      -0.037710  0.347151  0.559639 -0.280158  0.042335  0.028822   16.00
28      -0.038500  0.642522 -0.183891 -0.277464  0.182687  0.152665   33.00
29       0.014462  0.002951  0.294638 -0.395070  0.081461  0.024220   12.99
...           ...       ...       ...       ...       ...       ...     ...
284777 -0.002063  0.001344  0.262183 -0.105327 -0.022363 -0.060283    1.00
284778  0.141759  0.587119 -0.200998  0.267337 -0.152951 -0.065285   80.00
284779 -0.032129  0.768379  0.477688 -0.031833  0.014151 -0.066542   25.00
284780 -0.204280  1.158185  0.627801 -0.399981  0.510818  0.233265   30.00
284781 -0.147249  0.212931  0.354257 -0.241068 -0.161717 -0.149188   13.00
284782  0.235172 -0.681794 -0.668894  0.044657 -0.066751 -0.072447   12.82
284783  0.084783  0.721269 -0.529906 -0.240117  0.129126 -0.080620   11.46
284784 -0.373023  0.651122  1.073823  0.844590 -0.286676 -0.187719   40.00
284785  0.108519  0.688519 -0.460220  0.161939  0.265368  0.090245    1.79
284786  0.588482  0.632444 -0.201064  0.199251  0.438657  0.172923    8.95
284787  0.293632  0.107812 -0.935586  1.138216  0.025271  0.255347    9.99
284788  0.416765  0.064819 -0.608337  0.268436 -0.028069 -0.041367    3.99
284789 -0.148093 -0.038712  0.010209 -0.362666  0.503092  0.229921   60.50
284790  0.300245  0.000607 -0.376379  0.128660 -0.015205 -0.021486    9.81
284791 -0.074513 -0.003988 -0.113149  0.280378 -0.077310  0.023079   20.32
284792 -0.059545  0.242669 -0.665424 -0.269869 -0.170579 -0.030692    3.99
284793  0.163002  0.726365 -0.058282 -0.191813  0.061858 -0.043716    4.99
284794  0.088485 -0.076790 -0.095833  0.132720 -0.028468  0.126494    0.89
284795  0.890675 -1.253276  1.786717  0.320763  2.090712  1.232864    9.87
284796 -0.042114 -0.053206  0.316403 -0.461441  0.018265 -0.041068   60.00
284797  0.279598  0.371441 -0.559238  0.113144  0.131507  0.081265    5.49
284798  0.251791  0.057688 -1.508368  0.144023  0.181205  0.215243   24.05
284799 -0.348929  0.745323  0.704545 -0.127579  0.454379  0.130308   79.99
284800  0.297930 -0.359769 -0.315610  0.201114 -0.080826 -0.075071    2.68
284801  0.050343  0.102800 -0.435870  0.124079  0.217940  0.068803    2.69
284802  1.014480 -0.509348  1.436807  0.250034  0.943651  0.823731    0.77
284803  0.012463 -1.016226 -0.606624 -0.395255  0.068472 -0.053527   24.79
284804 -0.037501  0.640134  0.265745 -0.087371  0.004455 -0.026561   67.88
284805 -0.163298  0.123205 -0.569159  0.546668  0.108821  0.104533   10.00
284806  0.376777  0.008797 -0.473649 -0.818267 -0.002415  0.013649  217.00

        Class
0           0
1           0
2           0
3           0
4           0
5           0
6           0
7           0
8           0
9           0
10          0
```

11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0
29	0
...	...
284777	0
284778	0
284779	0
284780	0
284781	0
284782	0
284783	0
284784	0
284785	0
284786	0
284787	0
284788	0
284789	0
284790	0
284791	0
284792	0
284793	0
284794	0
284795	0
284796	0
284797	0
284798	0
284799	0
284800	0
284801	0
284802	0
284803	0

```
284804       0
284805       0
284806       0

[284807 rows x 31 columns]
```

[17]:
```
sns.scatterplot(credit_pd['Amount'],credit_pd['Class'])
```

[17]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x20048fc8940>
```