Trabalho 03

November 29, 2021

Aluno

- Isaque Elcio de Souza (RA: 225310)

0.0.1 Detecção de outliers

[122]:
```
import pandas as pd
import numpy as np
from sklearn.neighbors import LocalOutlierFactor
from sklearn import svm
from sklearn.ensemble import IsolationForest
from functools import reduce
```

[76]:
```
df = pd.read_csv('../Trabalho03/dados/dados3.csv')
```

[77]:
```
df.head()
```

[77]:

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	-2.97	1.020	-2.340	3.460	1.630	0.157	-2.660	0.559	-5.27	1.960
1	4.30	-0.817	1.410	-2.160	0.673	0.870	-1.220	1.620	3.43	-0.771
2	-2.62	0.378	-1.010	1.430	-0.278	-0.384	0.613	-0.880	-2.14	0.465
3	2.38	-0.356	0.731	-1.250	0.391	0.362	-0.817	1.000	1.85	-0.260
4	1.87	-0.568	0.440	-0.856	0.401	0.576	-0.568	0.793	1.55	-0.412

0.0.2 Local Outlier Factor (desnsidade)

Os testes com o valor do n_neighbors apresentou pequenas variações no resultado.

O valor da contamination = 0.01 (1%) obiteve 10 anomalias (outliers).

[105]:
```
clf = LocalOutlierFactor(n_neighbors=100, contamination = 0.01)
resultLof = clf.fit_predict(df)
```

[106]:
```
lof_outliers = []
for index, i in np.ndenumerate(resultLof):
    if(i == -1):
       lof_outliers.append(index[0])
lof_outliers = np.array(lof_outliers)
print("Número de anomalias (outliers): ", np.size(lof_outliers))
print("Index dos elementos com Anomalias (outliers): ", lof_outliers)
```

```
Número de anomalias (outliers):  10
Index dos elementos com Anomalias (outliers):  [  0  40 223 358 521 554 664 712
753 832]
```

[127]:
```
# Dados com anomalias encontrados - Local Outlier Factor
df.loc[df.index.isin(lof_outliers)]
```

[127]:

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	-2.97	1.020	-2.340	3.460	1.630	0.157	-2.660	0.559	-5.2700	1.9600
40	-2.94	1.190	-0.436	1.000	-0.932	-1.330	1.110	-1.380	-2.2600	0.7880
223	-1.49	-2.200	-1.580	2.200	0.765	2.470	0.926	-0.647	-0.8180	-1.6700
358	-1.06	-0.771	0.273	-1.490	-1.930	-0.709	3.400	-1.430	2.4100	-1.0200
521	2.77	-0.145	1.000	-1.590	0.335	0.138	-0.969	1.140	2.1100	-0.1310
554	-3.68	-1.890	-4.730	1.190	0.696	0.306	-0.464	1.630	-0.6800	1.6600
664	-1.58	-2.540	-3.620	2.650	2.160	2.690	-1.600	1.510	-1.4700	-0.0707
712	-2.96	1.500	-0.697	0.928	-0.760	-1.680	0.464	-0.913	-2.5200	1.4200
753	-3.14	0.296	-0.858	-0.421	-1.830	-1.720	2.610	-1.340	0.0272	0.5530
832	-2.15	0.469	-1.350	-1.120	-1.200	-2.070	0.909	0.224	0.5270	1.6700

0.0.3 One-class SVM

O valor da contamination = 0.01 (1%) obiteve 7 anomalias (outliers).

[109]:
```
clf_svm = svm.OneClassSVM(nu=0.01, kernel="rbf").fit(df)
one_class_svm =  clf_svm.predict(df)
```

[110]:
```
svm_outlier = []
for index, i in np.ndenumerate(one_class_svm):
    if(i == -1):
      svm_outlier.append(index[0])
svm_outlier = np.array(svm_outlier)
print("Número de anomalias (outliers): ", np.size(svm_outlier))
print("Index dos elementos com Anomalias (outliers): ", svm_outlier)
```

```
Número de anomalias (outliers):  7
Index dos elementos com Anomalias (outliers):  [  0 288 358 554 712 832 867]
```

[126]:
```
# Dados com anomalias encontrados - svm
df.loc[df.index.isin(svm_outlier)]
```

[126]:

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	-2.97	1.020	-2.340	3.460	1.630	0.1570	-2.660	0.559	-5.270	1.960
288	2.05	0.013	0.915	-1.200	0.157	0.0199	-0.615	0.704	1.520	-0.136
358	-1.06	-0.771	0.273	-1.490	-1.930	-0.7090	3.400	-1.430	2.410	-1.020
554	-3.68	-1.890	-4.730	1.190	0.696	0.3060	-0.464	1.630	-0.680	1.660
712	-2.96	1.500	-0.697	0.928	-0.760	-1.6800	0.464	-0.913	-2.520	1.420
832	-2.15	0.469	-1.350	-1.120	-1.200	-2.0700	0.909	0.224	0.527	1.670
867	4.86	-0.903	1.460	-2.480	0.833	0.9260	-1.570	2.010	3.860	-0.701

0.0.4 IsolationForest

O valor da contamination = 0.01 (1%) obiteve 10 anomalias (outliers).

[112]:
```
clf_IsolationForest = IsolationForest(n_estimators=100, contamination = 0.01,␣
↪random_state=42).fit(df)
IsolationForest_result = clf_IsolationForest.predict(df)
```

[113]:
```
IsolationForest_outlier = []
for index, i in np.ndenumerate(IsolationForest_result):
    if(i == -1):
      IsolationForest_outlier.append(index[0])
IsolationForest_outlier = np.array(IsolationForest_outlier)
print("Número de anomalias (outliers): ", np.size(IsolationForest_outlier))
print("Index dos elementos com Anomalias (outliers): ", IsolationForest_outlier)
```

```
Número de anomalias (outliers):  10
Index dos elementos com Anomalias (outliers):  [  0  40 223 358 554 664 712 753
816 832]
```

[125]:
```
# Dados com anomalias encontrados - Isolation Forest
df.loc[df.index.isin(IsolationForest_outlier)]
```

[125]:

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	-2.97	1.020	-2.340	3.460	1.630	0.157	-2.660	0.559	-5.2700	1.9600
40	-2.94	1.190	-0.436	1.000	-0.932	-1.330	1.110	-1.380	-2.2600	0.7880
223	-1.49	-2.200	-1.580	2.200	0.765	2.470	0.926	-0.647	-0.8180	-1.6700
358	-1.06	-0.771	0.273	-1.490	-1.930	-0.709	3.400	-1.430	2.4100	-1.0200
554	-3.68	-1.890	-4.730	1.190	0.696	0.306	-0.464	1.630	-0.6800	1.6600
664	-1.58	-2.540	-3.620	2.650	2.160	2.690	-1.600	1.510	-1.4700	-0.0707
712	-2.96	1.500	-0.697	0.928	-0.760	-1.680	0.464	-0.913	-2.5200	1.4200
753	-3.14	0.296	-0.858	-0.421	-1.830	-1.720	2.610	-1.340	0.0272	0.5530
816	-2.17	-0.310	-1.470	1.500	0.213	0.287	0.175	-0.313	-1.7000	0.2820
832	-2.15	0.469	-1.350	-1.120	-1.200	-2.070	0.909	0.224	0.5270	1.6700

0.0.5 Intersecção dos resultados de cada algoritmo (Combinação)

[124]:
```
resultado = reduce(np.intersect1d,(lof_outliers, svm_outlier,␣
↪IsolationForest_outlier))
df.loc[df.index.isin(resultado)]
```

[124]:

	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
0	-2.97	1.020	-2.340	3.460	1.630	0.157	-2.660	0.559	-5.270	1.96
358	-1.06	-0.771	0.273	-1.490	-1.930	-0.709	3.400	-1.430	2.410	-1.02
554	-3.68	-1.890	-4.730	1.190	0.696	0.306	-0.464	1.630	-0.680	1.66
712	-2.96	1.500	-0.697	0.928	-0.760	-1.680	0.464	-0.913	-2.520	1.42
832	-2.15	0.469	-1.350	-1.120	-1.200	-2.070	0.909	0.224	0.527	1.67