pd7-spytek-mikolaj

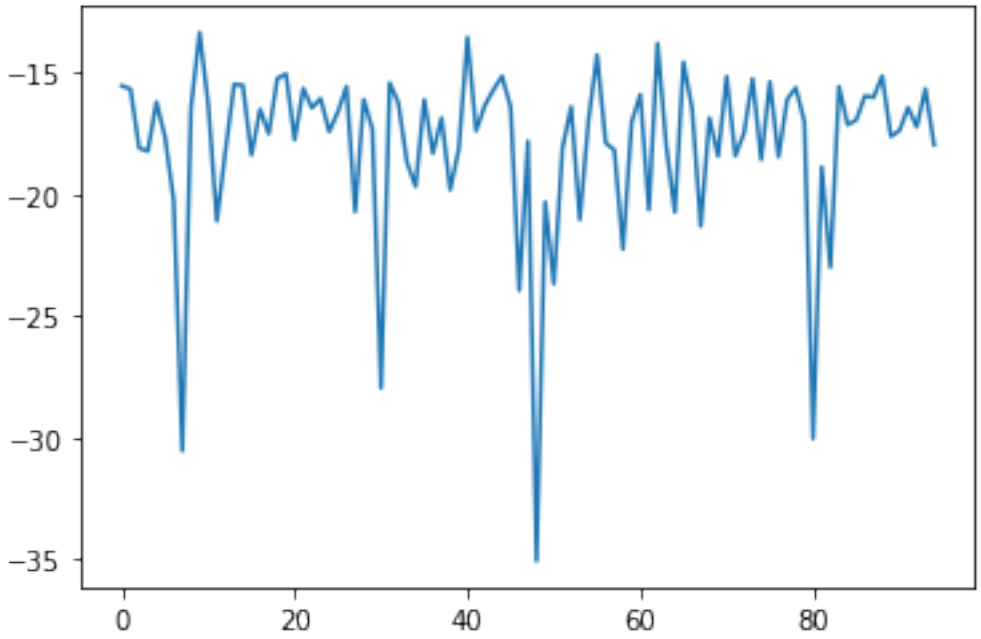
June 14, 2021

1 WUM - pd7

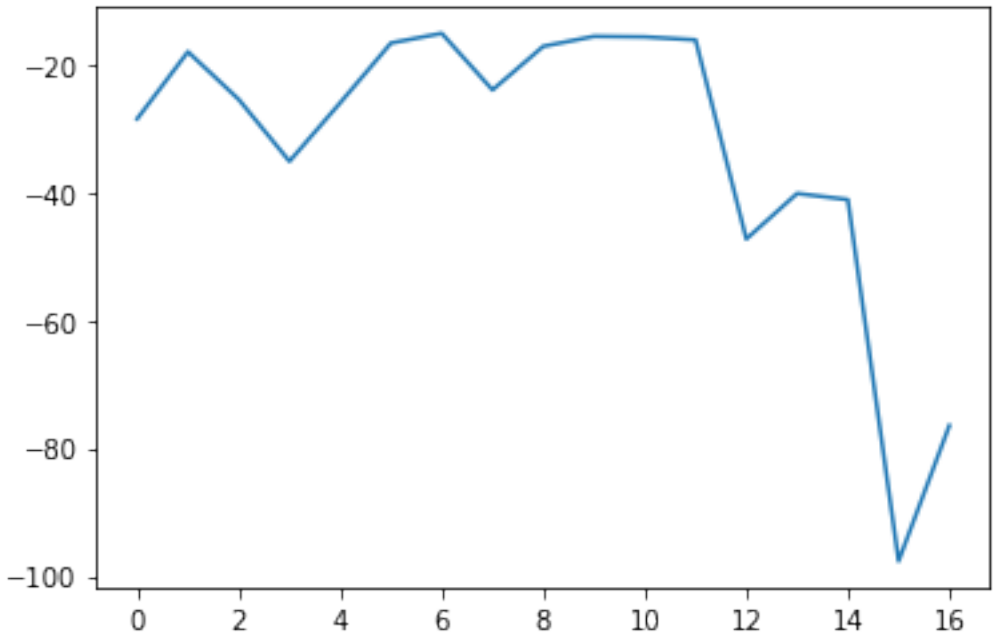
Mikołaj Spytek

[1]:
```
import pandas as pd
from sklearn.mixture import GaussianMixture
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt

scaler = StandardScaler()

train = pd.read_csv("train.csv")
test = pd.read_csv("test.csv")

test_true = test["class"]
test = test.drop(["class"], axis=1)

# train = pd.DataFrame(scaler.fit_transform(train), columns=train.columns)
# test = pd.DataFrame(scaler.fit_transform(test), columns=test.columns)


gm = GaussianMixture(n_components=2, max_iter=100000, covariance_type='tied',␣
↪random_state=42)

gm.fit(train)

test_pred = gm.predict(test)
gm.means_


probabilities_train = gm.score_samples(train)
probabilities_test = gm.score_samples(test)
```

[2]:
```
plt.plot(probabilities_train)
plt.show()
```



[3]:
```
plt.plot(probabilities_test)
plt.show()
```



[4]:
```
# wybieramy jakiś punkt odcięcia na podstawie 1-wszego wykresu
threshold = -35
prediction = [0 if i > threshold else 1 for i in probabilities_test ]
```

[5]:
```
prediction
```

[5]: [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1]

[6]:
```
test_true
```

[6]:
```
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    1
13    1
14    1
15    1
16    1
Name: class, dtype: int64
```

[7]:
```
from sklearn.metrics import precision_score, recall_score, f1_score


print("Precision: {}".format(precision_score(test_true, prediction)))
print("Recall: {}".format(recall_score(test_true, prediction)))
print("F1: {}".format(f1_score(test_true, prediction)))
```

```
Precision: 0.8333333333333334
Recall: 1.0
F1: 0.9090909090909091
```

1.0.1 Wnioski

Okazuje się, że algorytm GMM można wykorzystywać do detekcji outlierów