anomaly_detection

January 3, 2021

1 Anomaly Detection using Multivariate Gaussian Distribution

[ ]:

```
import pandas as pd
import numpy as np
import seaborn as sns

from scipy.stats import multivariate_normal
from sklearn.metrics import f1_score

import matplotlib.pyplot as plt
%matplotlib inline

train_set=pd.read_csv(r"Financial/train_data.csv")
test_set=pd.read_csv(r"Financial/test_data_hidden.csv")

from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
train_data=sc.fit_transform(train_set.iloc[:,0:-1])
test_data=sc.transform(test_data.iloc[:,0:-1])
test_target=test_data.iloc[:,-1]
```

[106]:

```
#function to calculate mean and covariance matrix of the features
def estimate_gaussian(dataset):
    mu = np.mean(dataset, axis=0)
    sigma = np.cov(dataset.T)
    return mu, sigma


#function to calculate the gaussian distribution probability of he data set
def multivariate_gaussian(dataset, mu, sigma):
    p = multivariate_normal(mean=mu, cov=sigma)
    return p.pdf(dataset)
```

[109]:

```
# creating function to find appropriate threshold
def select_threshold(probs, test_data):
    best_epsilon = 0
    best_f1 = 0
    f = 0
    stepsize = (max(probs) - min(probs)) / 1000;
    epsilons = np.arange(min(probs), max(probs), stepsize)
    for epsilon in np.nditer(epsilons):
        predictions = (probs < epsilon)
        f = f1_score(test_data, predictions, average='binary')
        if f > best_f1:
            best_f1 = f
            best_epsilon = epsilon
    return best_f1, best_epsilon

mu, sigma = estimate_gaussian(train_data)
p = multivariate_gaussian(train_data,mu,sigma)
```