

Anomaly Detection

```
In [1]: import matplotlib.pyplot as plt
import numpy as np
from numpy import genfromtxt
from scipy.stats import multivariate_normal
```

```
In [2]: def read_dataset(filePath,delimiter=','):
        return genfromtxt(filePath, delimiter=delimiter)

tr_data = read_dataset('anomaly_detect_data.csv')
```

```
In [3]: tr_data[:5,:]
```

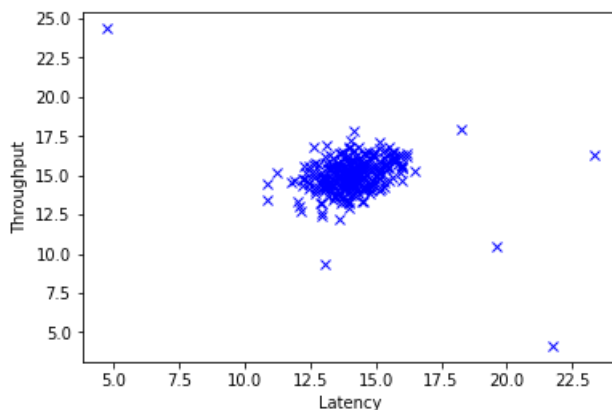
```
Out[3]: array([[13.047, 14.741],
               [13.409, 13.763],
               [14.196, 15.853],
               [14.915, 16.174],
               [13.577, 14.043]])
```

```
In [4]: n_training_samples = tr_data.shape[0]
n_dim = tr_data.shape[1]

print('Number of datapoints in training set: %d' % n_training_samples)
print('Number of dimensions/features: %d' % n_dim)
```

```
Number of datapoints in training set: 307
Number of dimensions/features: 2
```

```
In [5]: plt.xlabel('Latency')
plt.ylabel('Throughput')
plt.plot(tr_data[:,0],tr_data[:,1],'bx')
plt.show()
```



```
In [6]: def estimateGaussian(dataset):
        mu = np.mean(dataset, axis=0) # mean along each dimension / column
        sigma = np.cov(dataset.T)
        return mu, sigma

def multivariateGaussian(dataset,mu,sigma):
    p = multivariate_normal(mean=mu, cov=sigma)
    return p.pdf(dataset)
```

```
In [7]: mu, sigma = estimateGaussian(tr_data)
p = multivariateGaussian(tr_data,mu,sigma)
```

```
In [8]: thresh = 9e-05

# Determining Outliers/Anomalies
outliers = np.asarray(np.where(p < thresh))
outliers
```

```
Out[8]: array([[300, 301, 303, 304, 305, 306]], dtype=int64)
```

```
In [9]: plt.figure()
plt.xlabel('Latency')
plt.ylabel('Throughput')
plt.plot(tr_data[:,0],tr_data[:,1],'bx')
plt.plot(tr_data[outliers,0],tr_data[outliers,1],'ro')
plt.show()
```

