---

title: “Anomaly Detection” author: “SRIHARSHA SURINENI” date: “March 28, 2017” output: html_document —

Anomaly Detection

The objective of this analysis is to implement an anomaly detection algorithm to detect anomalous behavior in server computers. The features measure the through- put (mb/s), latency (ms) andother parameters of response of each server. While servers were operating, 1100 examples of how they were behaving were collected. Initial 1000 observations are all normal responses and there are 10 anomalies in final 100 observations which are used for validation of the model. Gaussian model will be used to detect anomalous observations in the dataset.

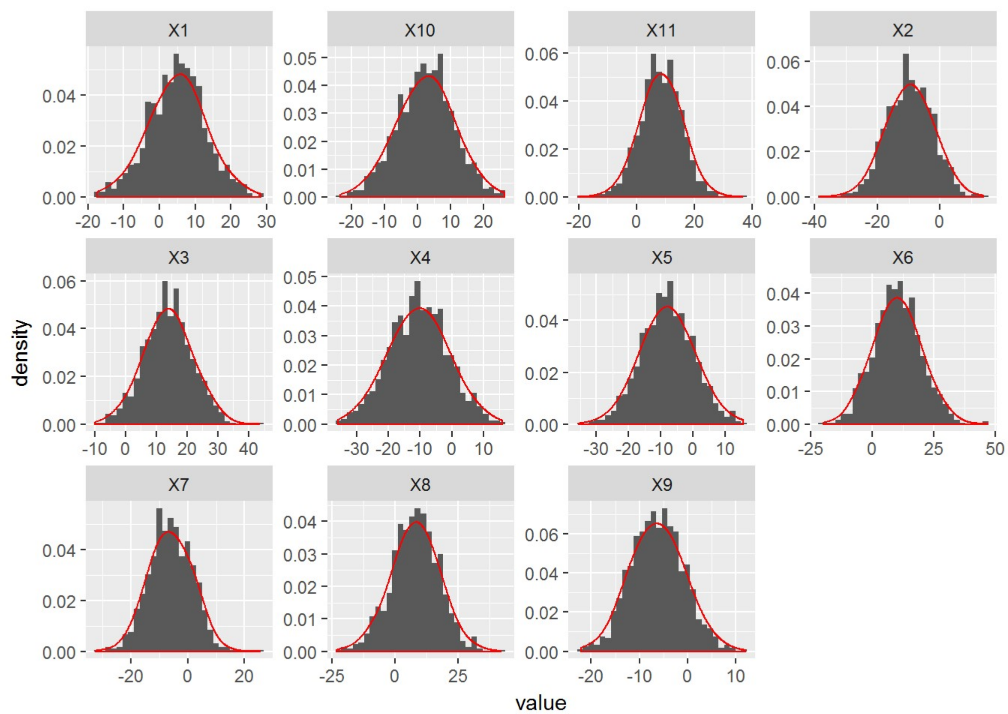
Model:

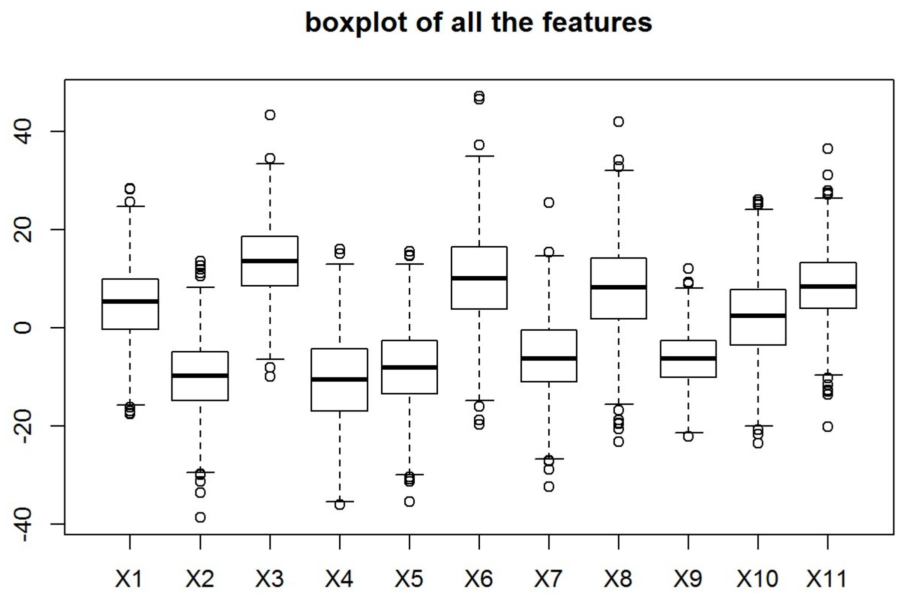
```
Generally, anomalies are rare occurances and it is very difficult to train
```

any supervised macine learning model effeciently as it requires approprite number of all the classes. In such scenarios, an alternative method of detecting anomalies by estimating joint probability distribution of

Exploratory analysis of the features:

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

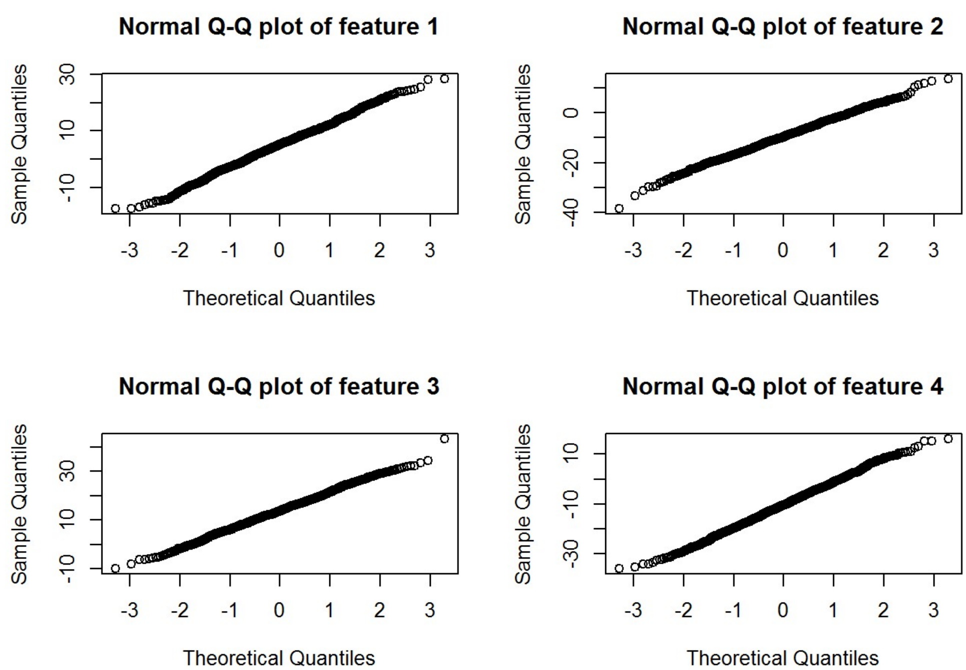




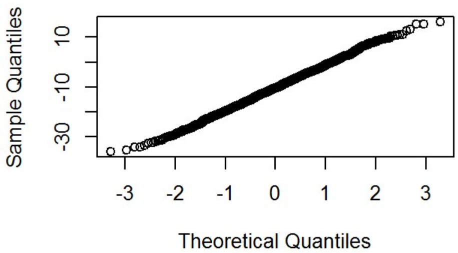
Train data:

Number of observations in train-data: `r length()

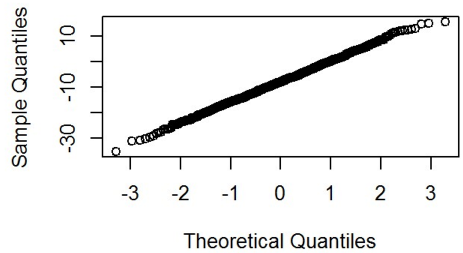
All the eleven features of the dataset look like they follow normal distribution from the above plots. Let’s have a look at their QQPlots and resuts of tests of normality:



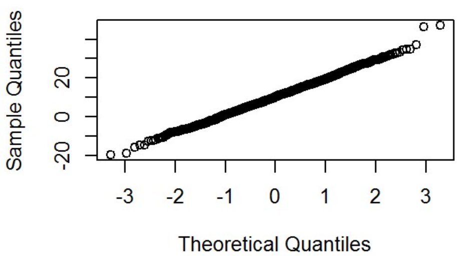
Normal Q-Q plot of feature 4



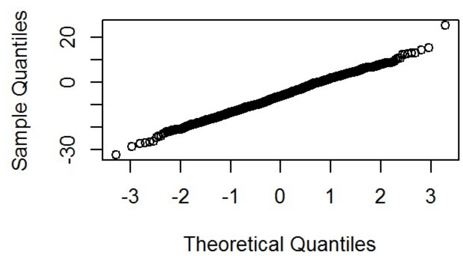
Normal Q-Q plot of feature 5



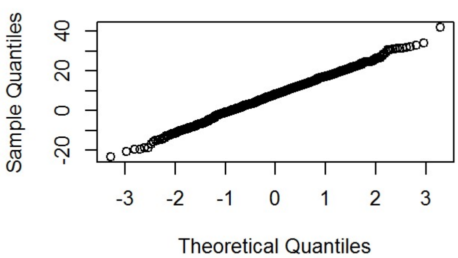
Normal Q-Q plot of feature 6



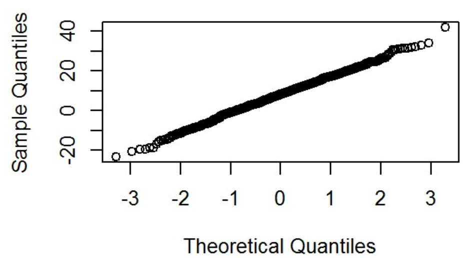
Normal Q-Q plot of feature 7



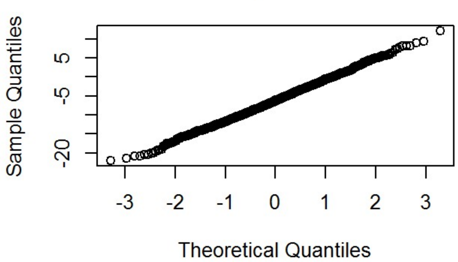
Normal Q-Q plot of feature 8



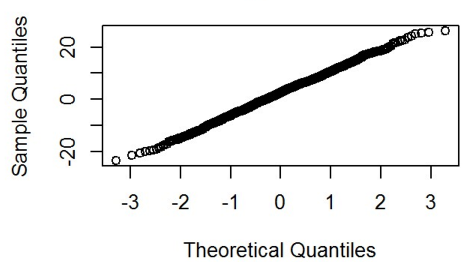
Normal Q-Q plot of feature 8

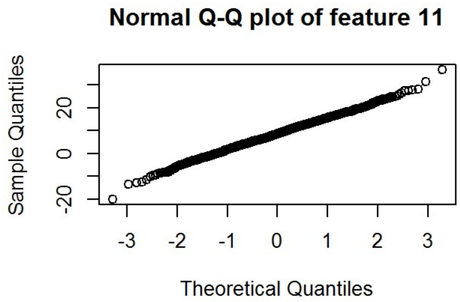


Normal Q-Q plot of feature 9



Normal Q-Q plot of feature 10





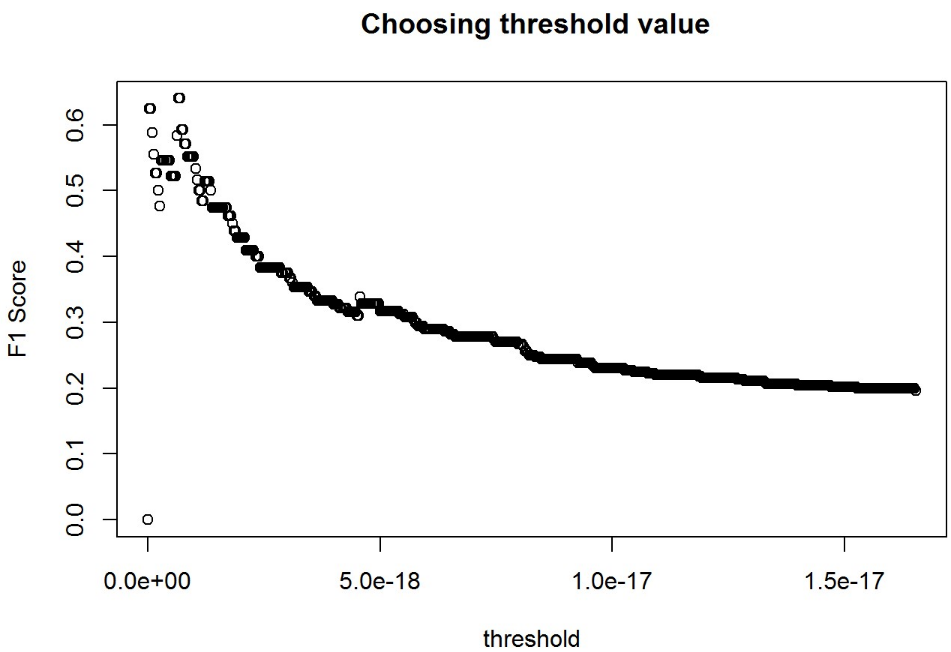
p-values of the Shapiro and Anderson_darling tests are as follows:

```
##                X1        X2        X3        X4        X5        X6
## shapiro 0.1609494 0.6697672 0.6582598 0.5722965 0.9618786 0.4980186
## adtest  0.1207252 0.8379918 0.7837818 0.9635222 0.9756475 0.9267057
##                X7        X8        X9       X10       X11
## shapiro 0.1774801 0.3881551 0.8004087 0.6326609 0.4665709
## adtest  0.1469380 0.2843801 0.7784174 0.5199585 0.6737168
```

There is no evidence from these tests as well to suggest any deviance from our assumption of normality of the features. So, our basic assumption that the features are derived from normal distribution is valid. Anomaly detection algorithm models the joint probability distributon function as multivariate normal distribution:

Under the modeled probability distribution function, joint probabilities for the observations of validation set are calculated and the threshold value probability to categorize anomalies, is chosen as the value which maximizes the performance criterion (F1- score) of this model on validation set. In this analysis, joint probability distribution has been modeled without any assumption of independence between the features (although it would have been very costly in terms of computing if we had a very large data set of aroung millions of rows). In case of large data sets, joint distribution can be modeled under assumtion of independence of features.

Choosing threshold:



Results:

From the above analysis, the threshold of joint probabiltiy dstribution function value to categorise an anomaly is 6.624241210^{-19}.

Misclassifcation table for this threshold for validation set:

83, 7, 2, 8

F1 - score for this threshold:

0.64

Conclusion:

This unsupervised machine learning technique is particularly useful, when there are very few anomalies in our data (such as 10 anomalies out of 1000 or 10000 observations), in which case applying other suervised machine learning techniques becomes difficult.

Disadvantages:

```
Major disadvantage of this model is it cant properly differentiate between rarely occuring normal cases and anoma
lies.
```

Reference: “Machine Learning Coursera Specialization by Mr. Andrew N G”

```
                                                ***THE END***
```