1. Confusion Matrix

The confusion matrix is used as an indication of the properties of a classification rule. It contains the number of elements that have been correctly or incorrectly classified for each class. We can see on its main diagonal the number of observations that have been correctly classified for each class; the off-diagonal elements indicate the number of observations that have been incorrectly classified. One benefit of a confusion matrix is that it is easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another).

For every instance in the test set, we compare the actual class to the class that was assigned by the trained classifier. A positive (negative) example that is correctly classified by the classifier is called a true positive (true negative); a positive (negative) example that is incorrectly classified is called a false negative (false positive). These numbers can be organized in a confusion matrix shown in Table below.

<table>
<thead>
<tr>
<th></th>
<th>Predicted negative</th>
<th>Predicted positive</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Negative Examples</strong></td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td><strong>Positive Examples</strong></td>
<td>C</td>
<td>D</td>
</tr>
</tbody>
</table>

Based on the values in Table 3.1, one can calculate all the measures defined above:
1. Accuracy is: \( \frac{a + d}{a + b + c + d} \)
2. Misclassification rate is: \( \frac{b + c}{a + b + c + d} \)
3. Precision is: \( \frac{d}{b + d} \)
4. True positive rate (Recall) is: \( \frac{d}{c + d} \)
5. False positive rate is: \( \frac{b}{a + b} \)
6. True negative rate (Specificity) is: \( \frac{a}{a + b} \)
7. False negative rate is: \( \frac{c}{c + d} \)
2. Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is most commonly used as dimensionality reduction technique in the pre-processing step for pattern-classification and machine learning applications. The goal is to project a dataset onto a lower-dimensional space with good class-separability in order avoid overfitting (“curse of dimensionality”) and also reduce computational costs.

LDA is closely related to analysis of variance (ANOVA) and regression analysis, which also attempt to express one dependent variable as a linear combination of other features or measurements. However, ANOVA uses categorical independent variables and a continuous dependent variable, whereas discriminant analysis has continuous independent variables and a categorical dependent variable (i.e. the class label, sensitive, malicious, best effort). Logistic regression and probit regression are more similar to LDA than ANOVA is, as they also explain a categorical variable by the values of continuous independent variables. These other methods are preferable in applications where it is not reasonable to assume that the independent variables are normally distributed, which is a fundamental assumption of the LDA method.

So, in a nutshell, often the goal of an LDA is to project a feature space (a dataset n-dimensional samples) onto a smaller subspace k (where k≤n−1) while maintaining the class-discriminatory information. In general, dimensionality reduction does not only help reducing
computational costs for a given classification task, but it can also be helpful to avoid overfitting by minimizing the error in parameter estimation (“curse of dimensionality”).

3. Algorithm of LDA

1. Compute the d-dimensional mean vectors for the different classes from the dataset.

2. Compute the scatter matrices (in-between-class and within-class scatter matrix).

3. Compute the eigenvectors (e1,e2,...,ed) and corresponding eigenvalues (λ1,λ2,...,λd) for the scatter matrices.

4. Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a d×k dimensional matrix W (where every column represents an eigenvector).

5. Use this d×k eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the matrix multiplication: Y=X×W (where X is a n×d dimensional matrix representing the n samples, and y are the transformed n×k-dimensional samples in the new subspace).

4. Classes of Traffic

In Traffic classification process LDA algorithm classifies network traffic according to various parameters (for example, based on port number or protocol) into a number of traffic classes described below. Each resulting traffic class can be treated differently in order to differentiate the service implied for the user (data generator/ consumer).

4.1 Sensitive traffic

Sensitive traffic is traffic the operator has an expectation to deliver on time. This includes VoIP, online gaming, video conferencing, and web browsing. Traffic management schemes are typically tailored in such a way that the quality of service of these selected uses is guaranteed, or at least prioritized over other classes of traffic. This can be accomplished by the absence of shaping for this traffic class, or by prioritizing sensitive traffic above other classes.
4.2 Best-effort traffic

Best effort traffic is all other kinds of non-detrimental traffic. This is traffic that the ISP deems isn't sensitive to Quality of Service metrics (jitter, packet loss, latency). A typical example would be peer-to-peer and email applications. Traffic management schemes are generally tailored so best-effort traffic gets what is left after sensitive traffic.

4.3 Undesired or Malicious traffic

This category is generally limited to the delivery of spam and traffic created by worms, botnets, and other malicious attacks. In some networks, this definition can include such traffic as non-local VoIP (for example, Skype) or video streaming services to protect the market for the 'in-house' services of the same type. In these cases, traffic classification mechanisms identify this traffic, allowing the network operator to either block this traffic entirely, or severely hamper its operation.

Figure 1: Auto correlation of Encoded cipher text showing on average 0.0065% of correlation between symbols

Measurement uncertainty and noise sometimes make it difficult to spot repeated patterns/behavior in a cipher text, even if such behavior is expected. The autocorrelation
sequence of a periodic signal has the same cyclic characteristics as the signal itself. Thus, autocorrelation can help verify the presence of cycles and determine their durations.

**Figure 2: Frequency distribution of 1KB of Cipher text**

Above figure 2 shows the frequency distribution displays the frequency of various alphabets in the cipher text. Each alphabet in the figure contains the frequency or count of the occurrences of values within a particular group or interval, and in this way, the table summarizes the distribution of values in the cipher text. Above figure shows the frequency distribution of 1KB of text showing almost equally distributed alphabets in the cipher text.

**Figure 3: Entropy of the Cipher text with its frequency**
Entropy is a measure of unpredictability of information content. It is usual in the computer industry to specify password strength in terms of information entropy, measured in bits, a concept from information theory. Instead of the number of guesses needed to find the password with certainty. In cryptanalysis, entropy is often roughly used as a measure of the unpredictability of a cryptographic key. The maximum entropy one can receive is around 8 bits i.e. all the bits in all the bytes of the cipher text are random. The average entropy of the system is around 7.25 and maximum entropy is around 7.5 bits.