| **FirstRun** | | **SecondRun** | | **Number of attributes** | | **Number of total instances** | | **Percent split** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |  | |  | |  | |  |
| **DataSet1** | 72.4 | | 70.7 | | 25 | | 30000 | | 90/10 |
| **DataSet2** | 91.6666666666667 | | 91.6666666666667 | | 8 | | 120 | | 90/10 |
| **DataSet3** | 75 | | 72.2222222222222 | | 33 | | 351 | | 90/10 |
| **DataSet4** | 74.6666666666667 | | 72 | | 5 | | 748 | | 90/10 |
| **DataSet5** | 96.4912280701754 | | 91.2280701754386 | | 32 | | 569 | | 90/10 |

Navie Baiysan result:

Neural Network Accuracy Summary:

| **FirstRun** | | **SecondRun** | | **Number of attributes** | | **Number of total instances** | | **Percent split** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |  | |  | |  | |  |
| **DataSet1** | 79.2333333333333 | | 77.4333333333333 | | 25 | | 30000 | | 90/10 |
| **DataSet2** | 77.8 | | 77.1333333333333 | | 8 | | 120 | | 90/10 |
| **DataSet3** | 71 | | 63.8888888888889 | | 33 | | 351 | | 90/10 |
| **DataSet4** | 76 | | 74.6666666666667 | | 5 | | 748 | | 90/10 |
| **DataSet5** | 64.9122807017544 | | 0 | | 32 | | 569 | | 90/10 |

Decision Tree:

| **FirstRun** | | **SecondRun** | | **Number of attributes** | | **Number of total instances** | | **Percent split** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |  | |  | |  | |  |
| **DataSet1** | 81.8 | | 80.3 | | 25 | | 30000 | | 90/10 |
| **DataSet2** | 100 | | 100 | | 8 | | 120 | | 90/10 |
| **DataSet3** | 86.1111111111111 | | 83.3333333333333 | | 33 | | 351 | | 90/10 |
| **DataSet4** | 78.6666666666667 | | 82.6666666666667 | | 5 | | 748 | | 90/10 |
| **DataSet5** | 96.4912280701754 | | 94.7368421052632 | | 32 | | 569 | | 90/10 |

Single layer perceptron

| **FirstRun** | | **SecondRun** | | **Number of attributes** | | **Number of total instances** | | **Percent split** | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | |  | |  | |  | |  |
| **DataSet1** | 78.2333333333333 | | 77.0666666666667 | | 25 | | 30000 | | 90/10 |
| **DataSet2** | 100 | | 75 | | 8 | | 120 | | 90/10 |
| **DataSet3** | 83.3333333333333 | | 83.3333333333333 | | 33 | | 351 | | 90/10 |
| **DataSet4** | 65.3333333333333 | | 80 | | 5 | | 748 | | 90/10 |
| **DataSet5** | 63.1578947368421 | | 38.5964912280702 | | 32 | | 569 | | 90/10 |

|  |  |
| --- | --- |
|  |  |



Report:

The model which performs best is SVM and decision tree /naïve bayes is worst in performance wise.

Even the best models sometimes perform poorly, and models with poor average perform best in certain dataset or selection of our testing data set and training dataset.

Conculsion:

I found that the all of algorithms achieved similar levels of classification accuracy given our feature space and dataset, making differentiation of them using standard evaluation metrics such as accuracy, and then it would be difficult.

I find that better differentiation of algorithms can be obtained by examining computational performance metrics such as build time and classification speed.

In comparing the classification speed, we find that Decision Tree, Naive Bayes, is able to identify network flows faster than the remaining algorithms. These are completed within few minutes.

Single Linear Classifier (Peceptron) and neural network took more time than DT, Naïve Bayes but faster than SVM. It took several minutes to build.

SVM took approx 30 min to build and calculate the accuracy of the all the five dataset.

**On the basis of performance:**

As there No such classifier suggests, it is universally best learning algorithm. Even the best models () perform poorly on some problems, and models that have poor average performance perform well on a few problems or dataset

***Decision Trees***

It creates a model based on a tree structure.

Nodes in the tree represent features, with branches representing possible values connecting features.

A leaf representing the class terminates a series of nodes and branches.

Determining the class of an instance is a matter of tracing the path of nodes and branches to the terminating leaf.

Cons:

***Single Layer Perceptron:***

It is simplest form of neural network and performs classification with only two classes. In the case pattern are drawn from two linearly separable classes.

Here two layer are there one is input layer and other is output layer. Hence neuron is the atomic computing unit of a neural network. From a very general outlook, it returns some output information from several input data.

Cons:

Functions that are not linearly separable eg: xor are not representable

***Neural Net:***

Multilayer neural network in which apart from input and output layer hidden layer is also present. We can change the number of hidden unit inside the particular hidden layer also we can change the number of hidden layer.

.