**PROJECT REPORT FOR CLASSIFICATION**

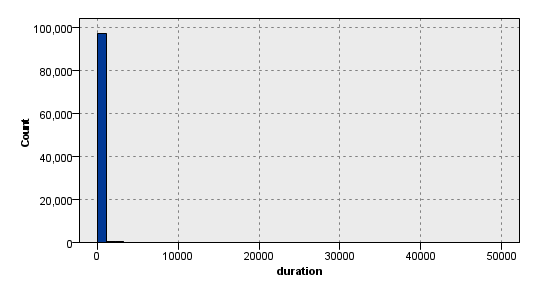
**Abstract**

It is important to classify the attack type of data from normal data to which can be useful to build a network intrusion detector. I worked on this data on classification techniques in developing a model to detect data intrusion in a network. Work file is a sample extracted from data files used for The Third International Knowledge Discovery and Data Mining Tools Competition, kddcupData. The sample file is a csv file and SPSS modeler 18.3 is used for the data analysis to classify the attacks. Analysis was done on 98327 records with 42 attributes. Of these huge data, concentration was on the attribute ‘connection\_type’ in which 22 type of attacks were present in the data set. Out of this, safe is the one with connection type ‘normal’ and remaining 21 are the records with unsafe attacks. With the given data, a model has been trained and developed for classifying any incoming data into the different attacks

**Analysis**

The sample ‘kddcupData’ file imported into the SPSS modeler. The data file is a csv file so ‘var.file’ node in the source tab has been used as a part of initiating the analysis. This node handles the comma delimited column text files. ‘Data audit’ node is attached to this for performing exploratory analysis. Upon running the data audit node, following shows the 42 fields of the data and their behavior.

**Duration** is the connection length, expressed as seconds. It is a continuous data ranging from 0 seconds to 42,448 seconds. Quality tab shows that there are 382 outliers and 376 extreme values.



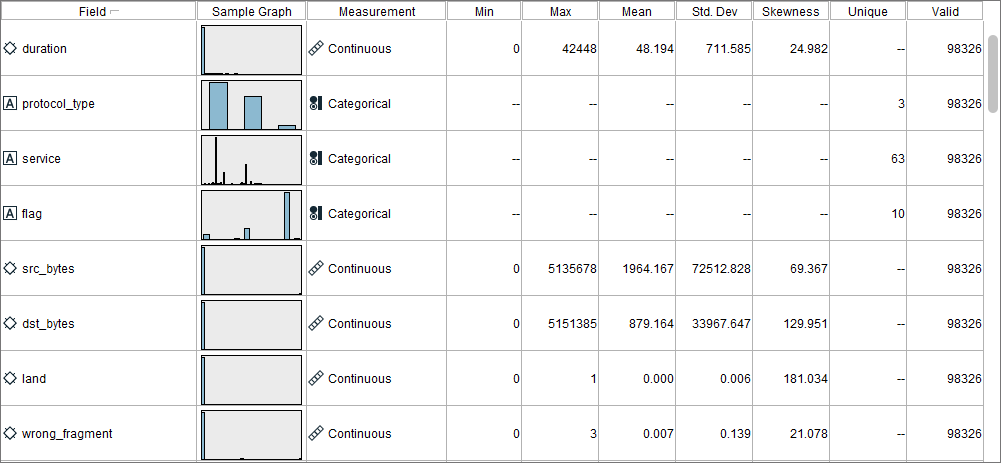
The above graph shows that most of the data points are under 10,000 seconds and negligible upto 43,000- this could be useful in retrieving any sample data for statistical analysis.

**Protocol\_type** field consists of categorical values icmp, tcp and udp. This field can be useful in considering as any connection types can be classified into these three data set categories.

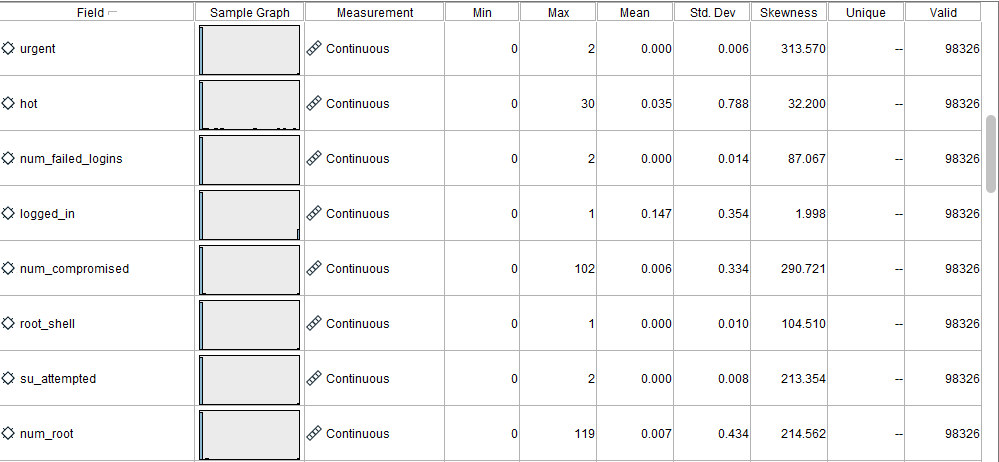
**Service** is also discrete with 63 types of network services on the destination. The histogram tells that most of network services are of type ecr\_i, http and private. While 90% of the data is distributed among these three network service types, only remaining 10% is shuffled between other 60 service types. Another field of this type is **flag** which denotes the normal or error status of the connection. Among 10 discrete values, 97% of the data points are covered under 3 categories REJ, S0 and SF. Remaining 3% out of 42,448 are distributed among other 7 categories. The significance of data sets distributed in low proportions is that the model will be easily trained and they can be easily classified to certain connection type which is similar to the existed data sample.

**Src\_bytes and dst\_bytes** hold the bytes from source to destination and destination to source respectively. They share a similar type of data sets with around 25 extreme values in each.

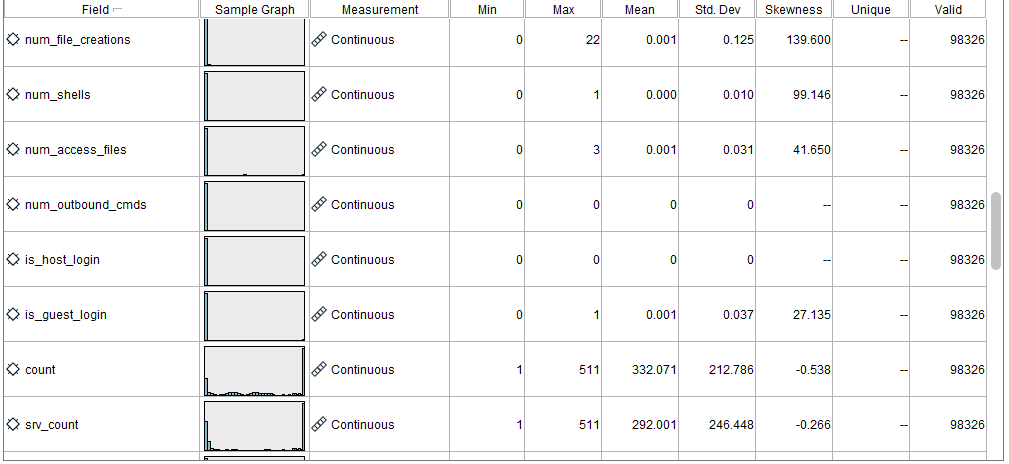
**Land, root\_shell** and **is\_guest\_login** holds discrete values 1 and 0. A common point among these is the three holds a large proportion for one value and other (0) is almost negligible where mean almost comes closer to 0.000. Mean of **num\_failed\_logins**, **su\_attempted** and **num\_access\_files** is also 0.000 where only few (<20) data points are other than 0 (1,2). Field **num\_shells** also hold same behavior but the data type of this field is continuous, different from previous ones. As they share a very negligible proportion under 0, the data points with 0 can be consider as extreme values. Only 1 data point is under 2 in **urgent,** denoting number of urgent packets. This could also be either significant factor or a data collection paradox. Another such is **wrong\_fragment** where 99.39 % of the data point in are covered under 0 but only remaining portion is under 1 and 3.

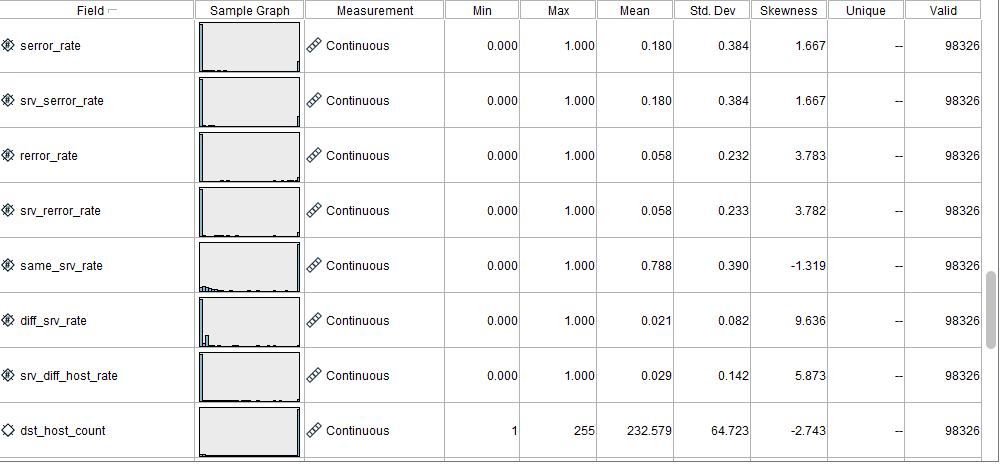


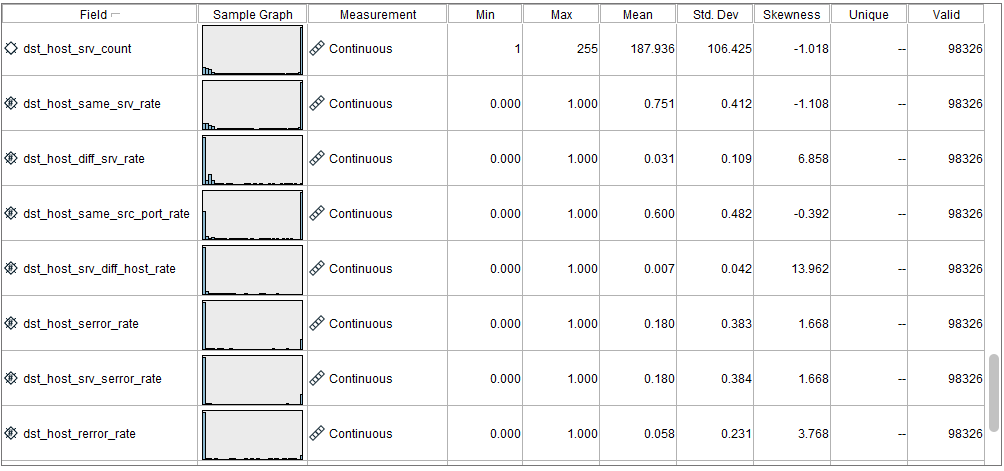
**Hot, num\_file\_creations, num\_compromised** and **num\_root** also have large proportion of concentration on one value and few (<1%) are distributed among other continuous values.

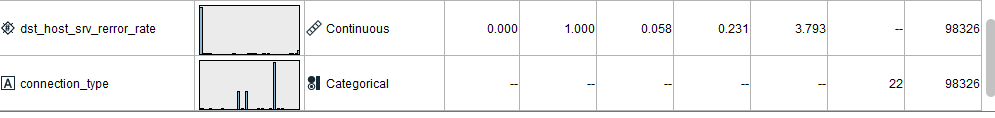


The attributes **num\_outbounds\_cmds** and **is\_host\_login** have constant values (zeroes) and hence can be removed from the analysis as they are of no significance in influencing the connection types.

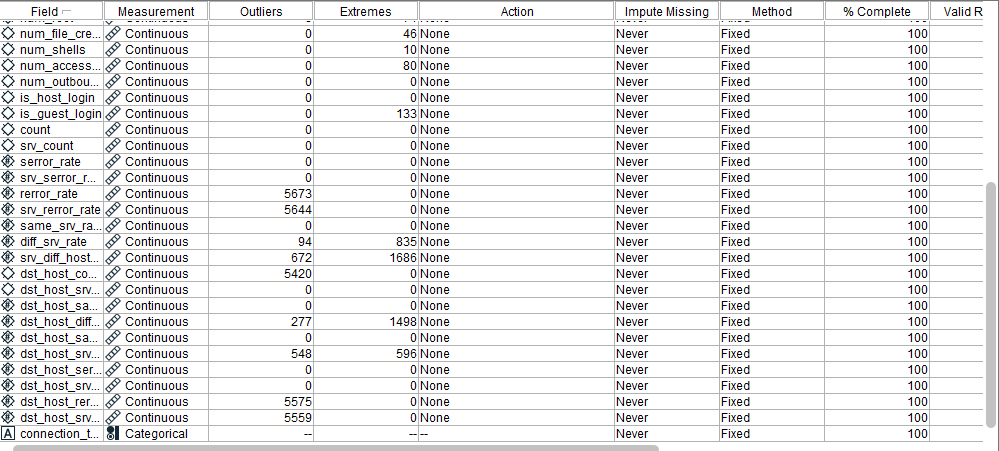






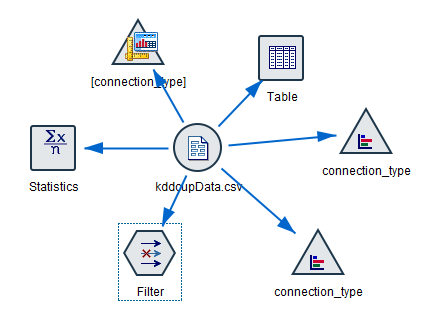


Remaining nodes are more or less same in their spread, though not distributed proportionately, they play a role in defining the connection types but some anomalies are seen in the attributes shown in below table

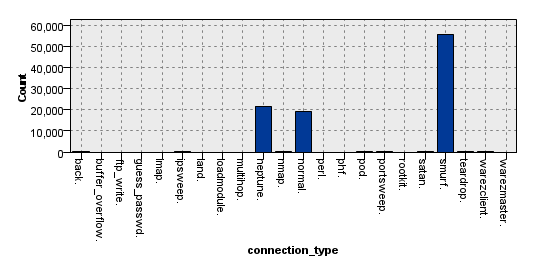


The target field ‘connection\_type’ includes 22 different categorical values. A more detailed view on this attribute is given in further discussions below.

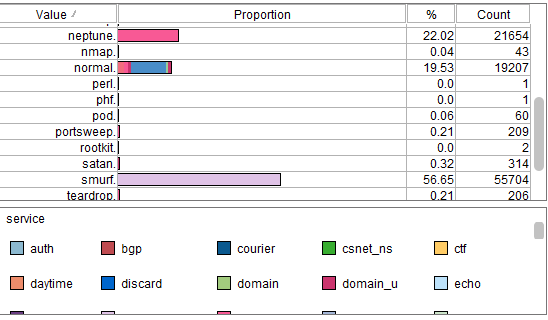
Furthermore, the following model was created in the SPSS modeler to evaluate the performances



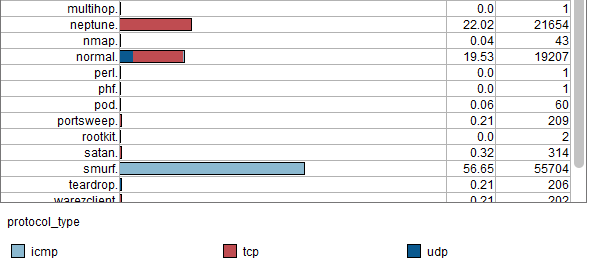
The connection type graphboard node shows that more data points are under connection type ‘smurf.’, ‘normal.’ and ‘neptune.’



The same is clearly depicted in the distribution node in graphs tab. The following is from this node when ‘connection type’ is overlaid by ‘service’



And the below is when ‘connection type’ is overlaid with ‘protocol\_type’

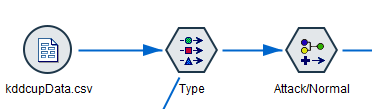


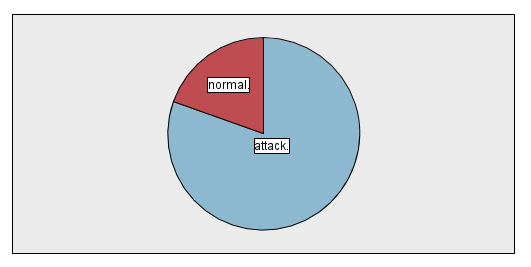
**Classification on connection types**

Based on the analysis, to discriminate good and bad connections, an attempt is made to build a model which could classify the connection types.

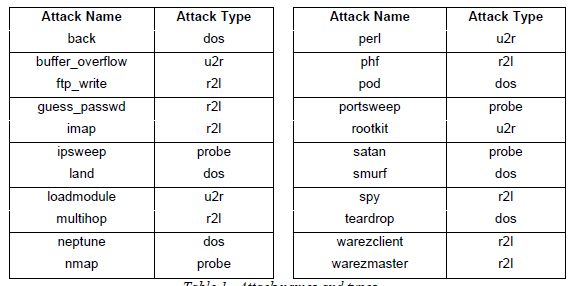
But before getting into deep analysis on classification, after importing the kddcupData.csv in to SPSS modeler, the connection types need to be reclassified to good (normal) and bad (attack) types. The ‘normal.’ in the connection type is reclassified to ‘normal’ and remaining types are reclassified into below two methods

1. In one method, the team has reclassified all the connection types to ‘normal’ and ‘attack’ as shown in below figures

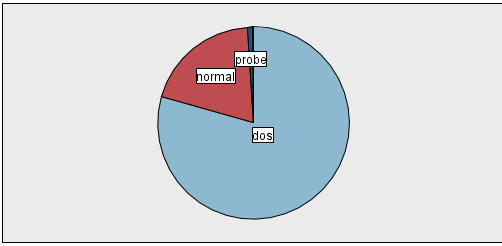




1. Another attempt is made in order to get more clearer view of connection types. Other than ‘normal’ type remaining are reclassified as shown in below table method, the connection types are reclassified to dos, r2l u2r and probe based on the below table



The below pi diagram gives a clear view on the connection types reclassified in this method.

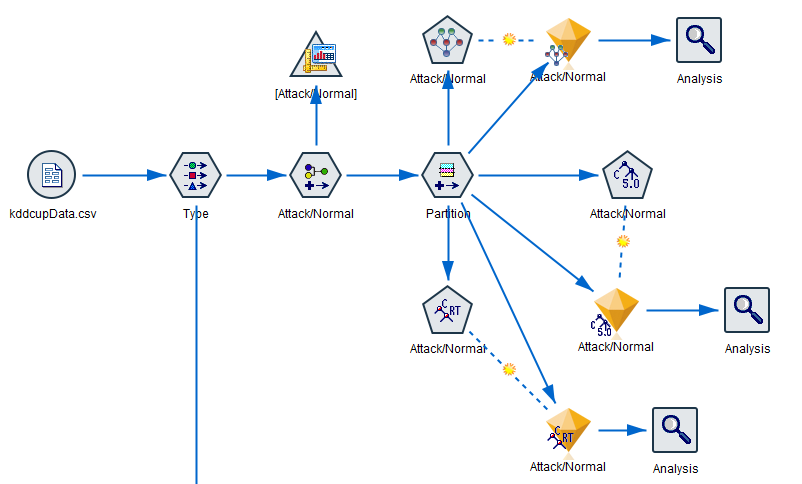


After reclassifying the target attribute, KNN node under Model tab is first used for classification but the model took minutes to execute and even after execution the team was unable to open the super node. The memory has also increased as shown in below picture and so this model was dropped off and the analysis over classification continued with other classification methods.

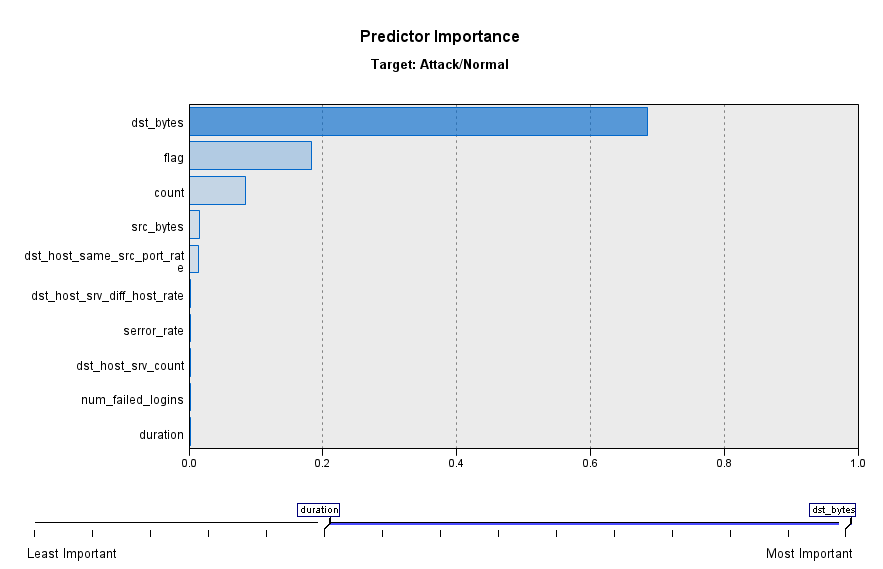


**Classification by method 1**

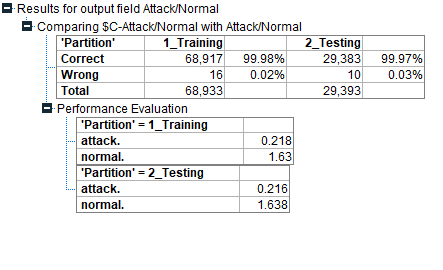
The reclassified data is subjected to partitioning with 70% data on training and 30% on testing. And the below is the model outlay; attached as ‘Project\_1\_classification.str’ file



The partitioned data is first analyzed with C5.0 classification node under modelling tab with reclassified attribute as a target and all other data fields as inputs. The execution resulted in many branches with below predictor importance histogram

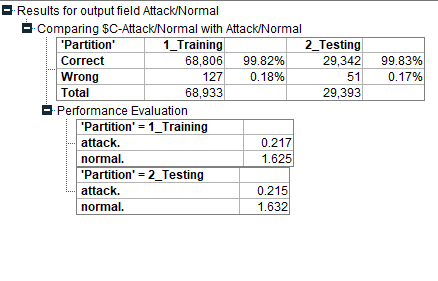


When analysis node in attached to C5 super node the following is resulted



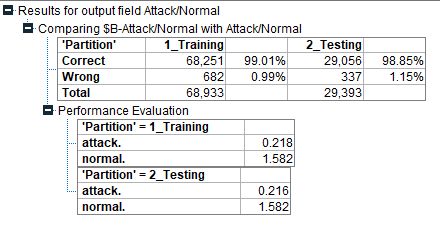
From the above the accuracy for test data is 99.97% which is nearer to training accuracy 99.98% (no overfitting)

By the histogram, the model is subjected to 85% of pruning with 30 records per child branch, the analysis node gave following



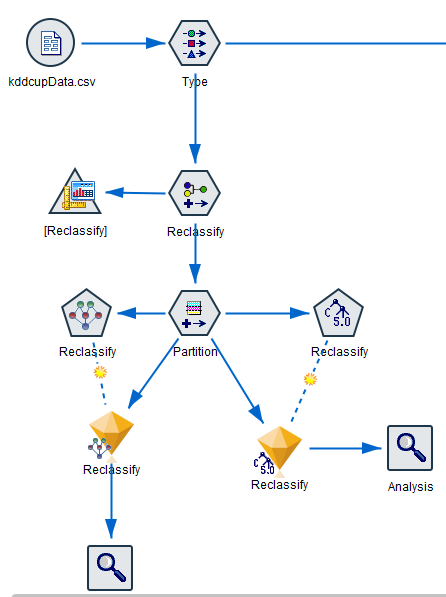
Which also resulted in good accuracy on testing data (which is now little greater than training data).

The whole model when executed with Bayesian network node, the testing accuracy is found to be a little low when compared to C5.0 classification

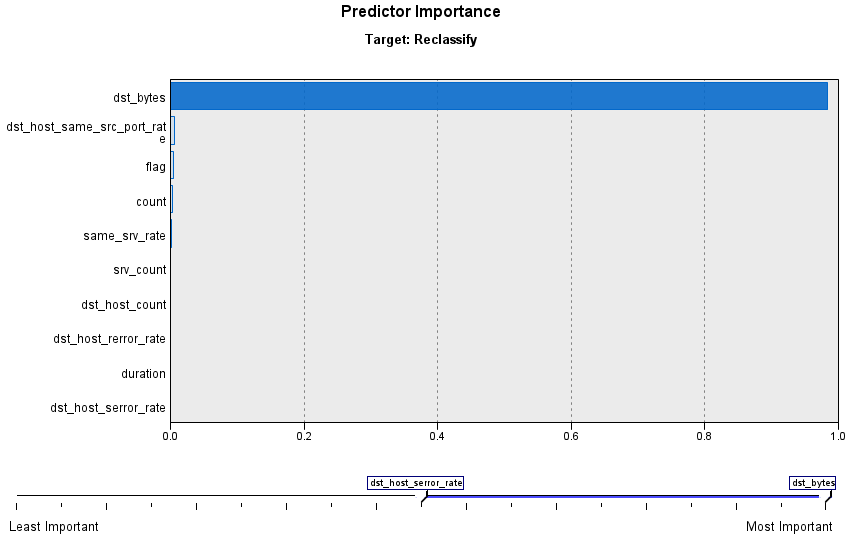


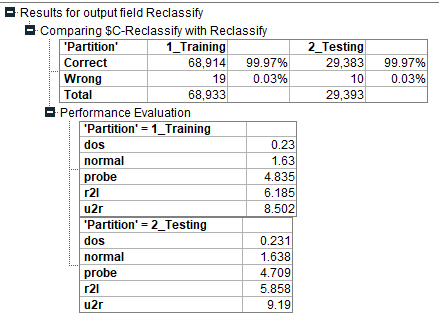
**Classification by method 2**

Below is model outlay on which analysis for normal and other 4 attacks was made; attached as ‘Project\_1\_classification.str’ file

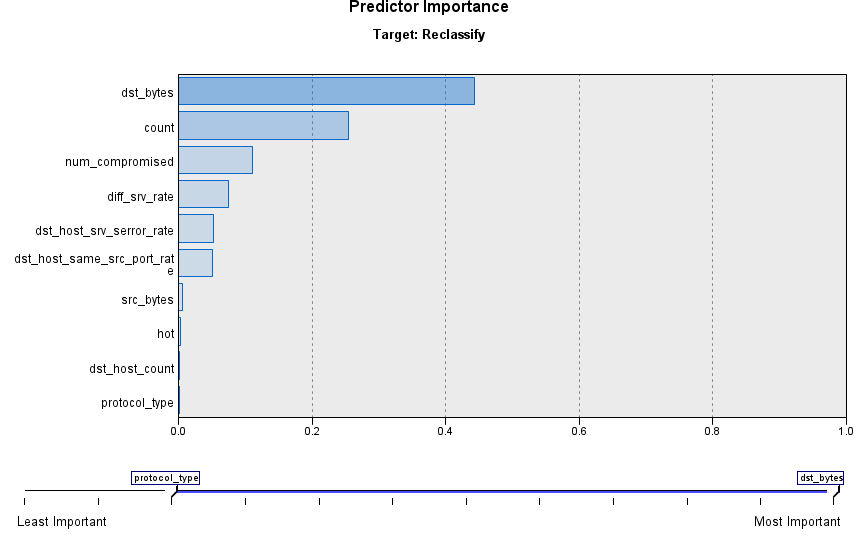


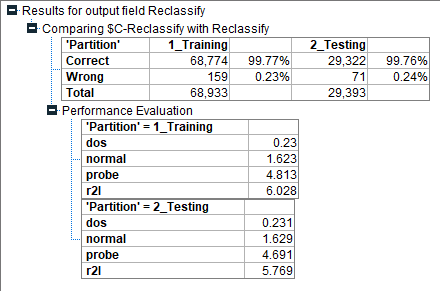
C5.0 classification node under modelling tab is executed with reclassified attribute as target and all other data fields as inputs. The execution resulted in many branches with below predictor importance histogram



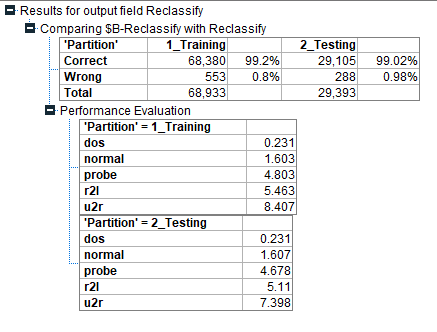


When compared to method 1, this has given a different range of predictor importance. Pruning this model very necessary. The model is hence subjected to 85% pruning with 30 records per child; resulted in



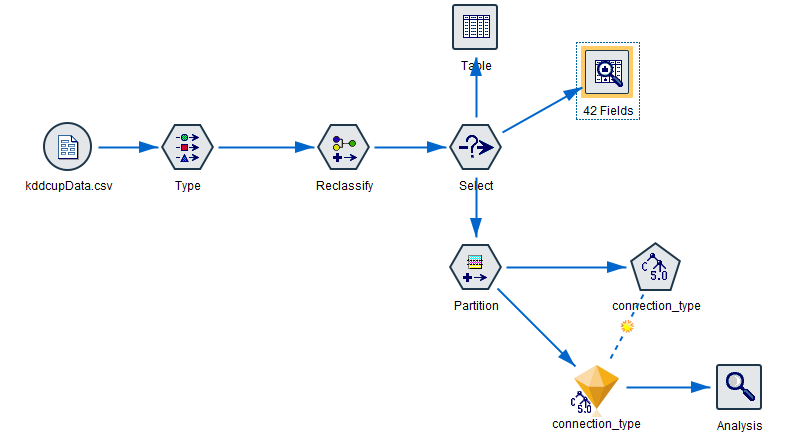


This model too executed for Bayesian network, which too has shown a little low accuracy when compared to C5.0 classification, depicted in the below figure



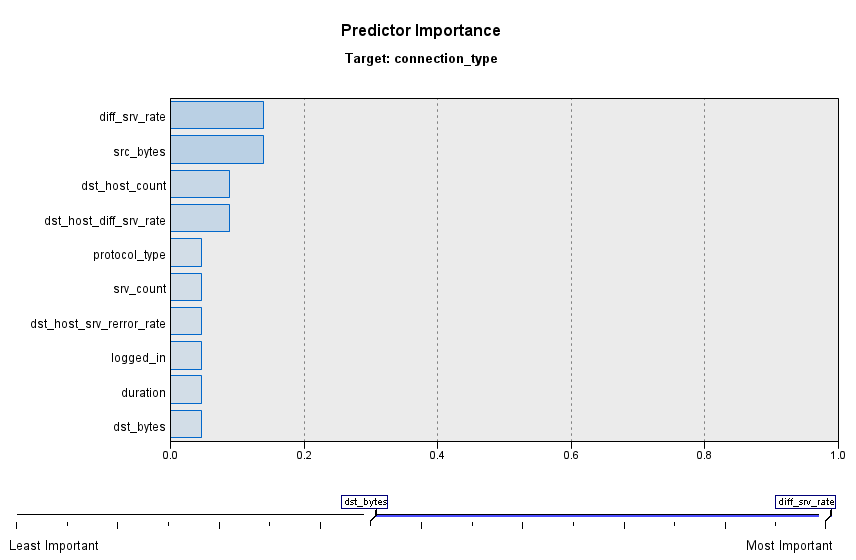
**Classification on attacks**

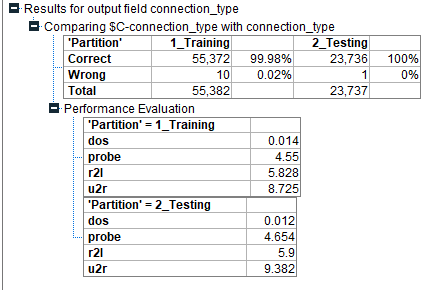
The team has attempted to characterize each of four types of attacks which were reclassified, above, to dos, r2l, u2r and probe. The following is the model structure created to classify the attack types and the same is attached as ‘Project\_1\_C5classification.str’ file



This model is completely analyzed for attacks and so the connection type ‘normal’ was discarded by select node. Then the data is partitioned for 70% training and 30% testing.

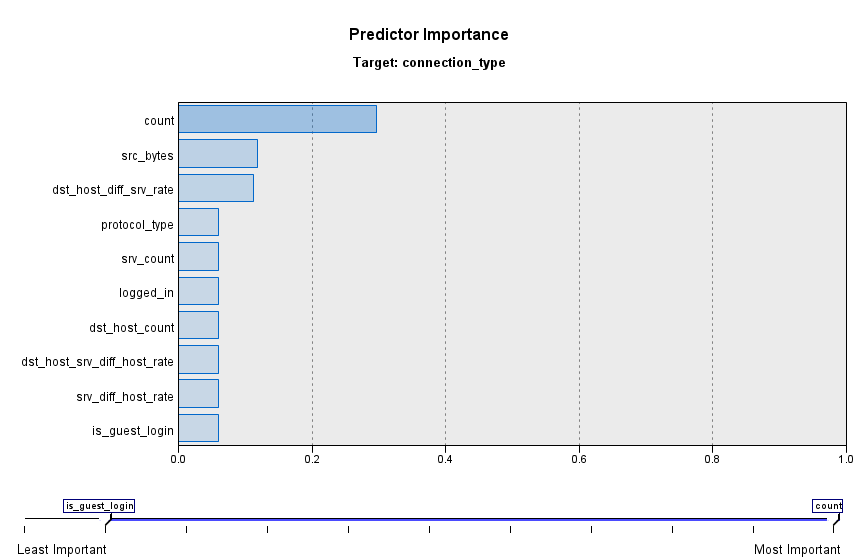
Upon attaching C5.0 node to this partitioned data gives following predictor importance

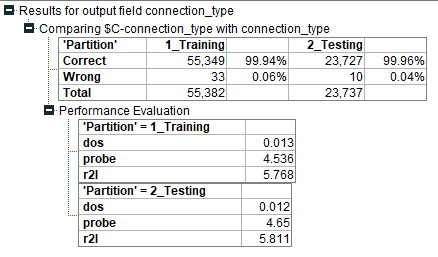




From the above results, there is a scope to prune the model

The following is the performance evaluations when the model is pruned to 75% with 10 records per child branch





**Conclusion**

The data is mostly polished with no abnormalities so the classification resulted in good accuracy. All the C5 decision trees can be viewed in individual str files. Though the u2r has few records, the performance evaluation is seen when the model is executed before pruning the data.