PROJECT REPORT FOR CLASSIFICATION

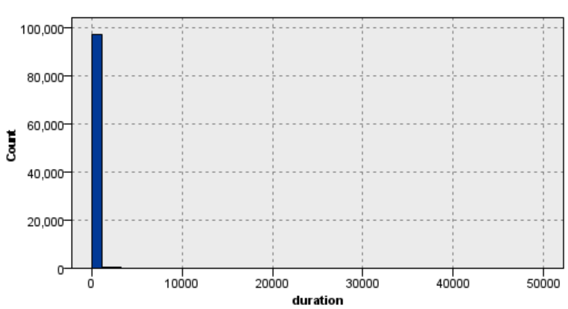
Abstract

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.

Field	Sample Graph	Measurement	Min	Max	Mean	Std. Dev	Skewness	Unique	Valid
duration		Continuous	0	42448	48.194	711.585	24.982	--	98326
protocol_type		Categorical	--	--	--	--	--	3	98326
service		Categorical	--	--	--	--	--	63	98326
flag		Categorical	--	--	--	--	--	10	98326
src_bytes		Continuous	0	5135678	1964.167	72512.828	69.367	--	98326
dst_bytes		Continuous	0	5151385	879.164	33967.647	129.951	--	98326
land		Continuous	0	1	0.000	0.006	181.034	--	98326
wrong_fragment		Continuous	0	3	0.007	0.139	21.078	--	98326

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.

Field	Sample Graph	Measurement	Min	Max	Mean	Std. Dev	Skewness	Unique	Valid
urgent		Continuous	0	2	0.000	0.006	313.570	--	98326
hot		Continuous	0	30	0.035	0.788	32.200	--	98326
num_failed_logins		Continuous	0	2	0.000	0.014	87.067	--	98326
logged_in		Continuous	0	1	0.147	0.354	1.998	--	98326
num_compromised		Continuous	0	102	0.006	0.334	290.721	--	98326
root_shell		Continuous	0	1	0.000	0.010	104.510	--	98326
su_attempted		Continuous	0	2	0.000	0.008	213.354	--	98326
num_root		Continuous	0	119	0.007	0.434	214.562	--	98326

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.

Field	Sample Graph	Measurement	Min	Max	Mean	Std. Dev	Skewness	Unique	Valid
num_file_creations		Continuous	0	22	0.001	0.125	139.600	--	98326
num_shells		Continuous	0	1	0.000	0.010	99.146	--	98326
num_access_files		Continuous	0	3	0.001	0.031	41.650	--	98326
num_outbound_cmds		Continuous	0	0	0	0	--	--	98326
is_host_login		Continuous	0	0	0	0	--	--	98326
is_guest_login		Continuous	0	1	0.001	0.037	27.135	--	98326
count		Continuous	1	511	332.071	212.786	-0.538	--	98326
srv_count		Continuous	1	511	292.001	246.448	-0.266	--	98326

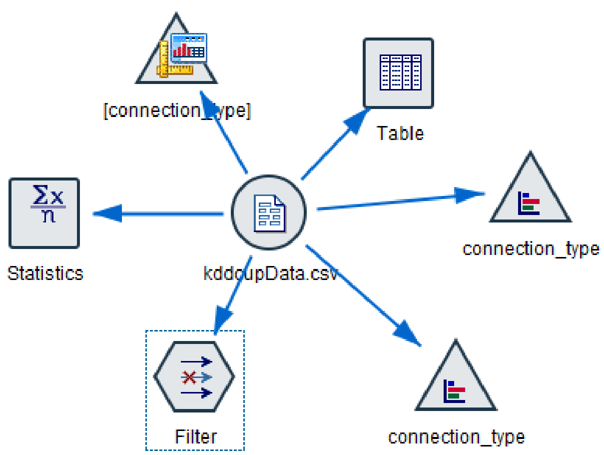
Field	Sample Graph	Measurement	Min	Max	Mean	Std. Dev	Skewness	Unique	Valid
serror_rate		Continuous	0.000	1.000	0.180	0.384	1.667	--	98326
srv_serror_rate		Continuous	0.000	1.000	0.180	0.384	1.667	--	98326
rerror_rate		Continuous	0.000	1.000	0.058	0.232	3.783	--	98326
srv_rerror_rate		Continuous	0.000	1.000	0.058	0.233	3.782	--	98326
same_srv_rate		Continuous	0.000	1.000	0.788	0.390	-1.319	--	98326
diff_srv_rate		Continuous	0.000	1.000	0.021	0.082	9.636	--	98326
srv_diff_host_rate		Continuous	0.000	1.000	0.029	0.142	5.873	--	98326
dst_host_count		Continuous	1	255	232.579	64.723	-2.743	--	98326
dst_host_srv_count		Continuous	1	255	187.936	106.425	-1.018	--	98326
dst_host_same_srv_rate		Continuous	0.000	1.000	0.751	0.412	-1.108	--	98326
dst_host_diff_srv_rate		Continuous	0.000	1.000	0.031	0.109	6.858	--	98326
dst_host_same_src_port_rate		Continuous	0.000	1.000	0.600	0.482	-0.392	--	98326
dst_host_srv_diff_host_rate		Continuous	0.000	1.000	0.007	0.042	13.962	--	98326
dst_host_serror_rate		Continuous	0.000	1.000	0.180	0.383	1.668	--	98326
dst_host_srv_serror_rate		Continuous	0.000	1.000	0.180	0.384	1.668	--	98326
dst_host_rerror_rate		Continuous	0.000	1.000	0.058	0.231	3.768	--	98326
dst_host_srv_rerror_rate		Continuous	0.000	1.000	0.058	0.231	3.793	--	98326
connection_type		Categorical	--	--	--	--	--	22	98326

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

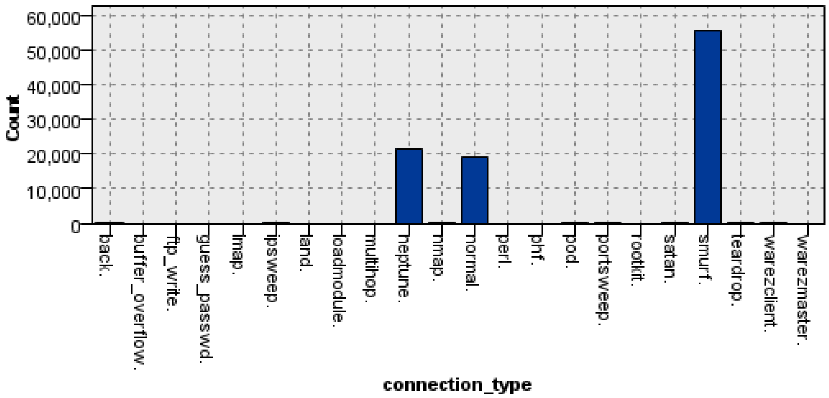
Field	Measurement	Outliers	Extremes	Action	Impute Missing	Method	% Complete	Valid R
num_file_cre...	Continuous	0	46	None	Never	Fixed	100	
num_shells	Continuous	0	10	None	Never	Fixed	100	
num_access...	Continuous	0	80	None	Never	Fixed	100	
num_outbou...	Continuous	0	0	None	Never	Fixed	100	
is_host_login	Continuous	0	0	None	Never	Fixed	100	
is_guest_login	Continuous	0	133	None	Never	Fixed	100	
count	Continuous	0	0	None	Never	Fixed	100	
srv_count	Continuous	0	0	None	Never	Fixed	100	
serror_rate	Continuous	0	0	None	Never	Fixed	100	
srv_serror_r...	Continuous	0	0	None	Never	Fixed	100	
rerror_rate	Continuous	5673	0	None	Never	Fixed	100	
srv_rerror_rate	Continuous	5644	0	None	Never	Fixed	100	
same_srv_ra...	Continuous	0	0	None	Never	Fixed	100	
diff_srv_rate	Continuous	94	835	None	Never	Fixed	100	
srv_diff_host...	Continuous	672	1686	None	Never	Fixed	100	
dst_host_co...	Continuous	5420	0	None	Never	Fixed	100	
dst_host_srv...	Continuous	0	0	None	Never	Fixed	100	
dst_host_sa...	Continuous	0	0	None	Never	Fixed	100	
dst_host_diff...	Continuous	277	1498	None	Never	Fixed	100	
dst_host_sa...	Continuous	0	0	None	Never	Fixed	100	
dst_host_srv...	Continuous	548	596	None	Never	Fixed	100	
dst_host_ser...	Continuous	0	0	None	Never	Fixed	100	
dst_host_srv...	Continuous	0	0	None	Never	Fixed	100	
dst_host_rer...	Continuous	5575	0	None	Never	Fixed	100	
dst_host_srv...	Continuous	5559	0	None	Never	Fixed	100	
connection_t...	Categorical	--	--	--	Never	Fixed	100	

The target field 'connection_type' includes 22 different categorical values. A more detailed view on this attribute is given in further discussions below.

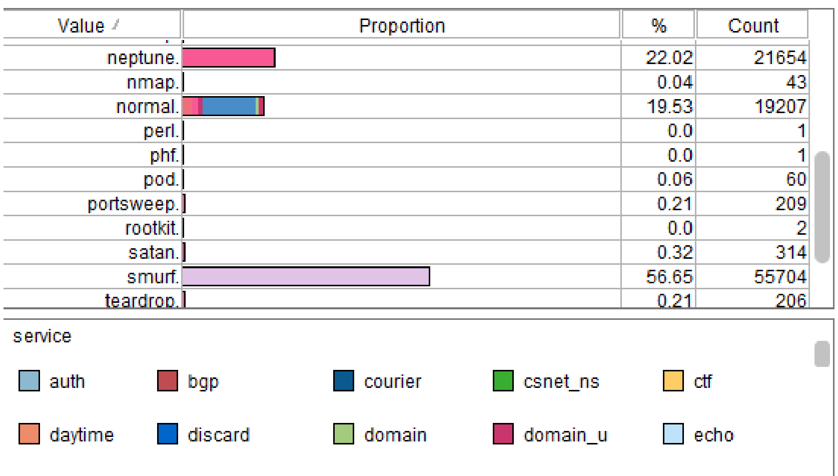
Furthermore, the following model was created in the SPSS modeler to evaluate the performances



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'

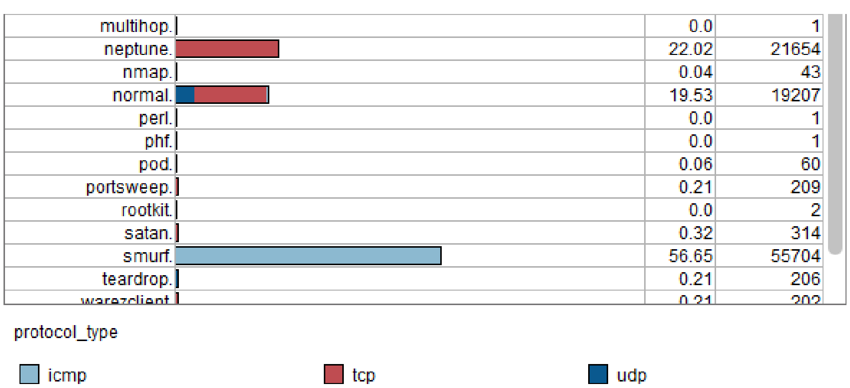


Value	Proportion	%	Count
neptune.		22.02	21654
nmap.		0.04	43
normal.		19.53	19207
perl.		0.0	1
phf.		0.0	1
pod.		0.06	60
portsweep.		0.21	209
rootkit.		0.0	2
satan.		0.32	314
smurf.		56.65	55704
teardrop.		0.21	206

service

auth, bgp, courier, csnet_ns, ctf, daytime, discard, domain, domain_u, echo

And the below is when 'connection type' is overlaid with 'protocol_type'



multihop.		0.0	1
neptune.		22.02	21654
nmap.		0.04	43
normal.		19.53	19207
perl.		0.0	1
phf.		0.0	1
pod.		0.06	60
portsweep.		0.21	209
rootkit.		0.0	2
satan.		0.32	314
smurf.		56.65	55704
teardrop.		0.21	206
warezclient.		0.21	202

protocol_type

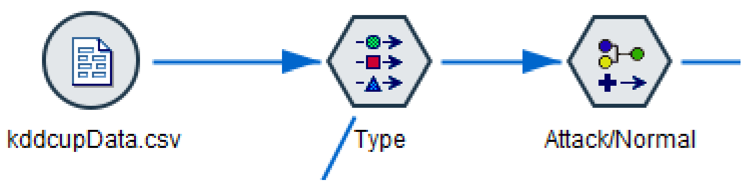
icmp, tcp, udp

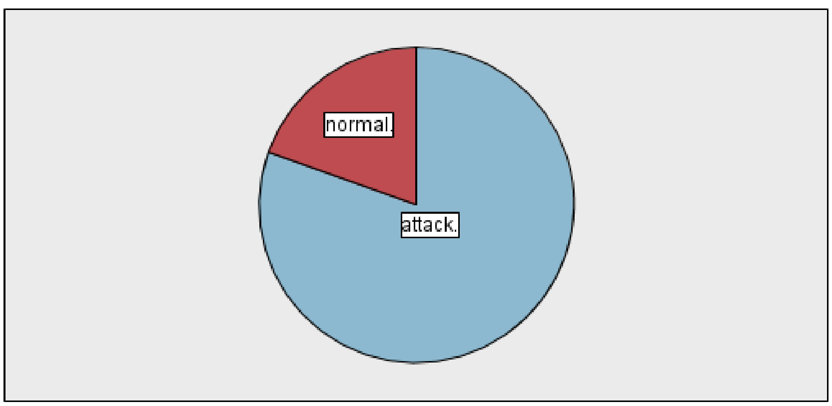
Classification on connection types

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

i) In one method, the team has reclassified all the connection types to 'normal' and 'attack' as shown in below figures

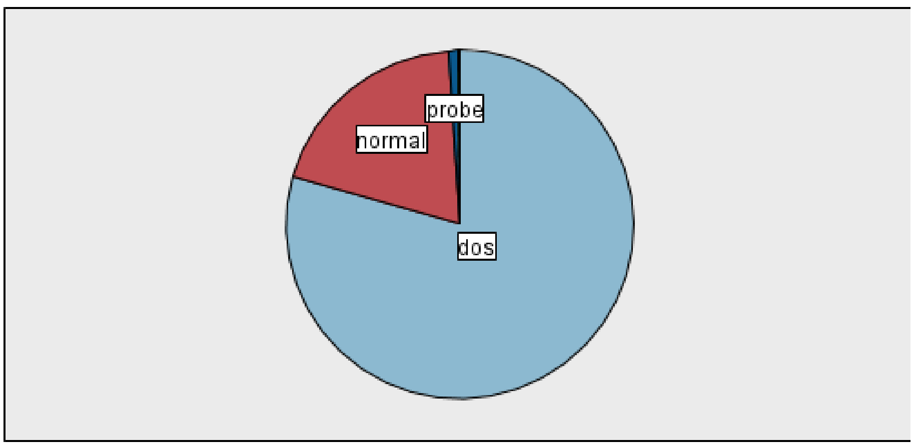




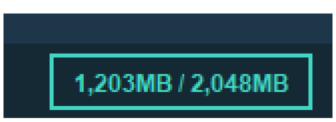
ii) Another attempt is made in order to get more clearer view of connection types. Other than 'normal' type remaining are reclassified as shown in below table method, the connection types are reclassified to dos, r2l u2r and probe based on the below table

Attack Name	Attack Type	Attack Name	Attack Type
back	dos	perl	u2r
buffer_overflow	u2r	phf	r2l
ftp_write	r2l	pod	dos
guess_passwd	r2l	portsweep	probe
imap	r2l	rootkit	u2r
ipsweep	probe	satan	probe
land	dos	smurf	dos
loadmodule	u2r	spy	r2l
multihop	r2l	teardrop	dos
neptune	dos	warezclient	r2l
nmap	probe	warezmaster	r2l

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

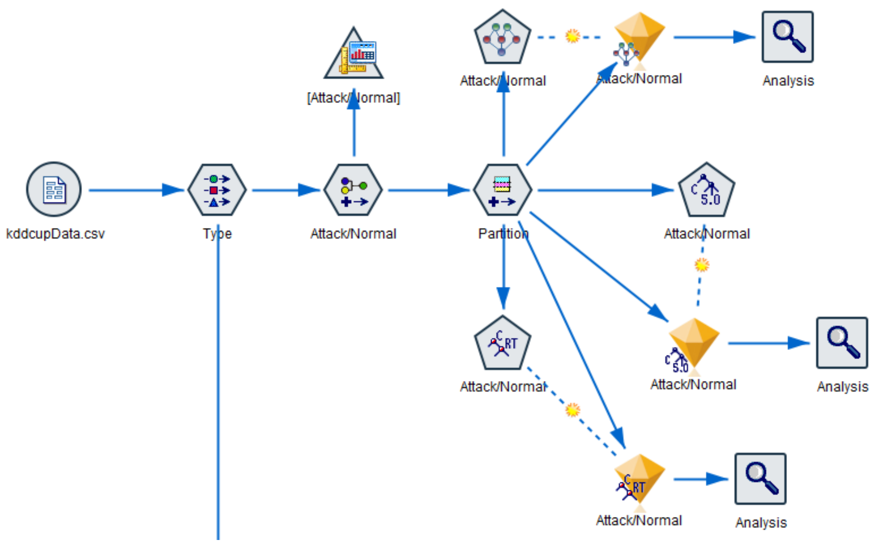


After reclassifying the target attribute, KNN node under Model tab is first used for classification but the model took minutes to execute and even after execution the team was unable to open the super node. The memory has also increased as shown in below picture and so this model was dropped off and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

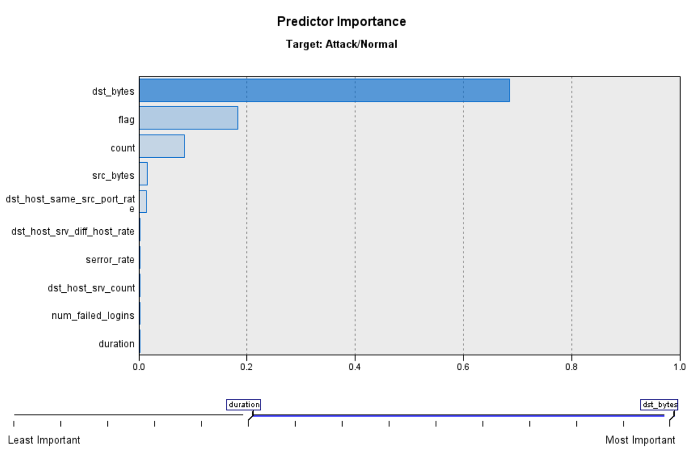


Classification by method 1

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

Results for output field Attack/Normal

Comparing $C-Attack/Normal with Attack/Normal

'Partition'	1_Training		2_Testing	
Correct	68,917	99.98%	29,383	99.97%
Wrong	16	0.02%	10	0.03%
Total	68,933		29,393	

Performance Evaluation

'Partition' = 1_Training	
attack.	0.218
normal.	1.63

'Partition' = 2_Testing	
attack.	0.216
normal.	1.638

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

Results for output field Attack/Normal

Comparing $C-Attack/Normal with Attack/Normal

'Partition'	1_Training		2_Testing	
Correct	68,806	99.82%	29,342	99.83%
Wrong	127	0.18%	51	0.17%
Total	68,933		29,393	

Performance Evaluation

'Partition' = 1_Training	
attack.	0.217
normal.	1.625

'Partition' = 2_Testing	
attack.	0.215
normal.	1.632

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

Results for output field Attack/Normal

Comparing $B-Attack/Normal with Attack/Normal

'Partition'	1_Training		2_Testing	
Correct	68,251	99.01%	29,056	98.85%
Wrong	682	0.99%	337	1.15%
Total	68,933		29,393	

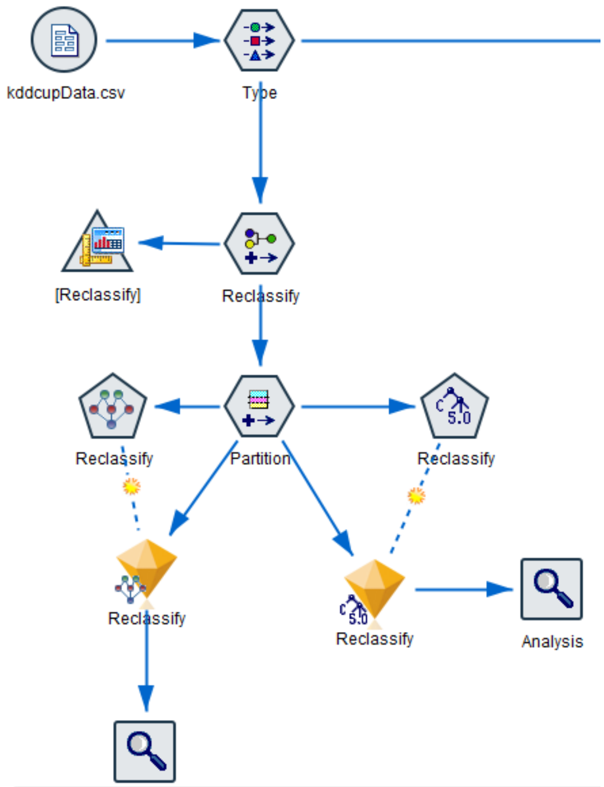
Performance Evaluation

'Partition' = 1_Training	
attack.	0.218
normal.	1.582

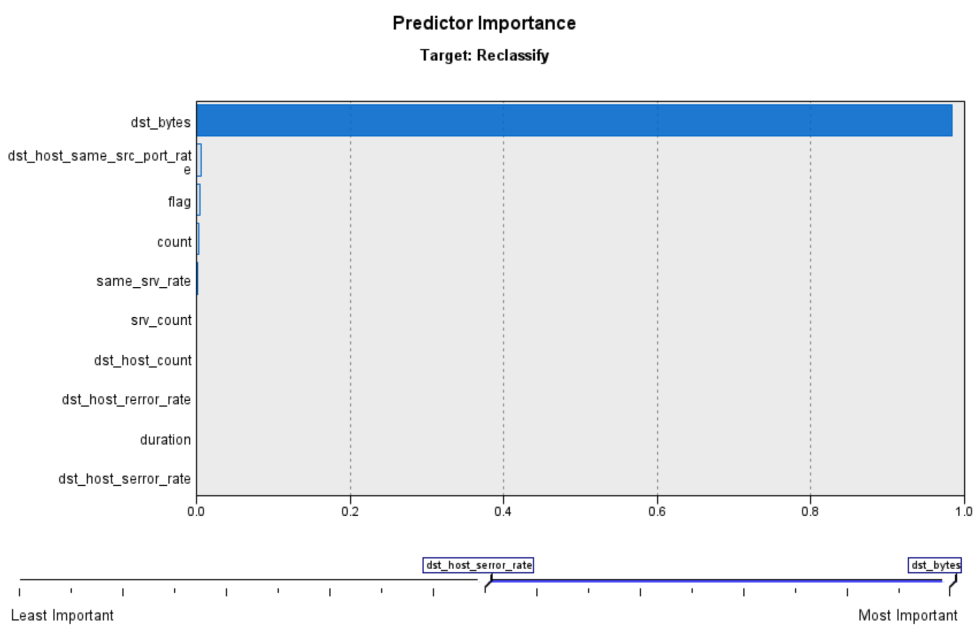
'Partition' = 2_Testing	
attack.	0.216
normal.	1.582

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



Results for output field Reclassify

Comparing $C-Reclassify with Reclassify

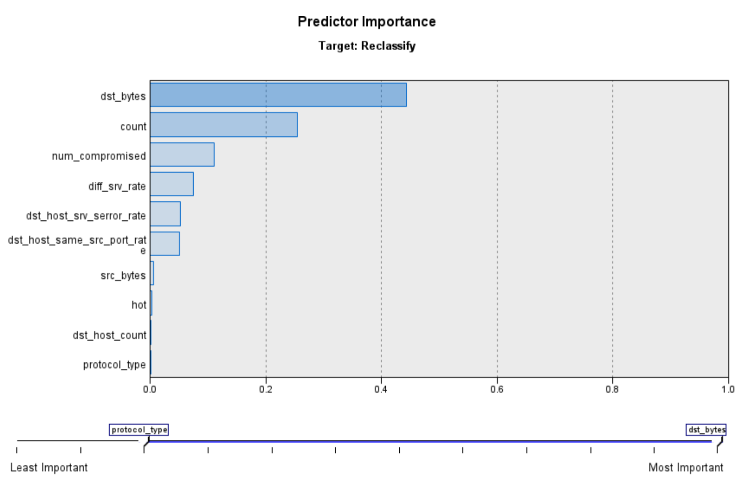
'Partition'	1_Training		2_Testing	
Correct	68,914	99.97%	29,383	99.97%
Wrong	19	0.03%	10	0.03%
Total	68,933		29,393	

Performance Evaluation

'Partition' = 1_Training	
dos	0.23
normal	1.63
probe	4.835
r2l	6.185
u2r	8.502

'Partition' = 2_Testing	
dos	0.231
normal	1.638
probe	4.709
r2l	5.858
u2r	9.19

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



Results for output field Reclassify

Comparing $C-Reclassify with Reclassify

'Partition'	1_Training		2_Testing	
Correct	68,774	99.77%	29,322	99.76%
Wrong	159	0.23%	71	0.24%
Total	68,933		29,393	

Performance Evaluation

'Partition' = 1_Training	
dos	0.23
normal	1.623
probe	4.813
r2l	6.028

'Partition' = 2_Testing	
dos	0.231
normal	1.629
probe	4.691
r2l	5.769

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

Results for output field Reclassify

Comparing $B-Reclassify with Reclassify

'Partition'	1_Training		2_Testing	
Correct	68,380	99.2%	29,105	99.02%
Wrong	553	0.8%	288	0.98%
Total	68,933		29,393	

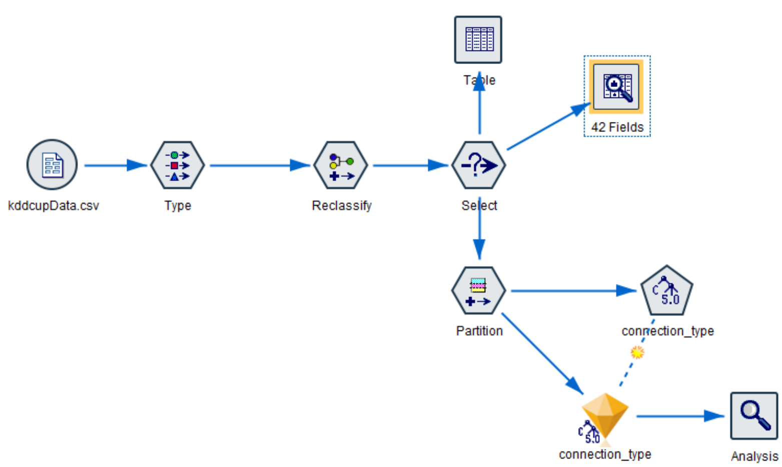
Performance Evaluation

'Partition' = 1_Training	
dos	0.231
normal	1.603
probe	4.803
r2l	5.463
u2r	8.407

'Partition' = 2_Testing	
dos	0.231
normal	1.607
probe	4.678
r2l	5.11
u2r	7.398

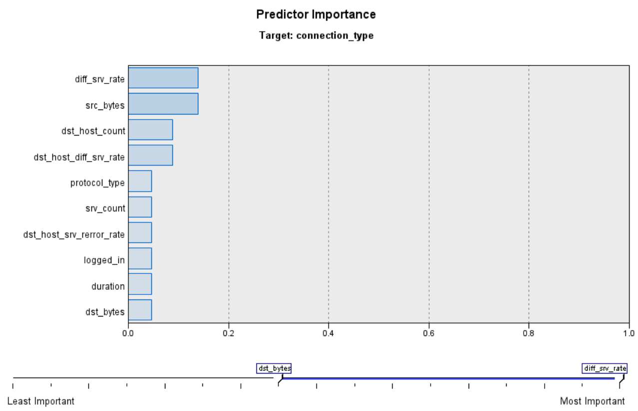
Classification on attacks

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



Results for output field connection_type

Comparing $C-connection_type with connection_type

'Partition'	1_Training		2_Testing	
Correct	55,372	99.98%	23,736	100%
Wrong	10	0.02%	1	0%
Total	55,382		23,737	

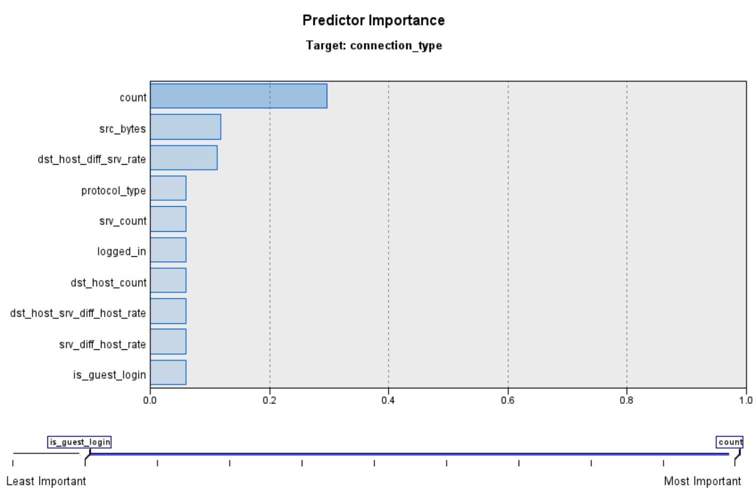
Performance Evaluation

'Partition' = 1_Training	
dos	0.014
probe	4.55
r2l	5.828
u2r	8.725

'Partition' = 2_Testing	
dos	0.012
probe	4.654
r2l	5.9
u2r	9.382

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



Results for output field connection_type

Comparing $C-connection_type with connection_type

'Partition'	1_Training		2_Testing	
Correct	55,349	99.94%	23,727	99.96%
Wrong	33	0.06%	10	0.04%
Total	55,382		23,737	

Performance Evaluation

'Partition' = 1_Training	
dos	0.013
probe	4.536
r2l	5.768

'Partition' = 2_Testing	
dos	0.012
probe	4.65
r2l	5.811

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.