PROJECT-SUMMARY

NETWORK INTRUSION DETECTION

MULTINOMIAL CLASSIFICATION

Problem statement- Task is to build network intrusion detection system to detect anamolies and attacks in the network.

Multinomial classification: Activity is normal or DOS or PROBE or R2L or U2R

Diagnostic- The business problem can be solved by classifying the activity as normal or in attack category.

Target variable - Activity category(NORMAL, DOS ,PROBE , R2L ,U2R)

Independent variables – Basic, content related and host based network features

Step1. Data preparation

(Data profiling, Missing values imputation, Outliers handling ,Feature encoding)

1. Data is loaded from Test.txt and Train.txt available in NSL_dataset
2. Variable *attack* in dataset is mapped to attack category and new variable is *'attack_cat'* and 'attack' variable is dropped
3. 9 variables are dropped based on DATA PROFILING
   'dst_host_same_srv_rate','dst_host_serror_rate','dst_host_srv_rerror_rate','dst_host_srv_serror_rate','num_outbound_cmds','num_root','srv_rerror_rate','srv_serror_rate', 'service'
4. No missing values are present. Numerical data is handled for outliers.
5. Scalar data is standardized using StandardScalor() from sklearn. Preproceesing module.
6. **FEATURE ENCODING** -one- hot encoding is performed on categorical variables. *protocol_type and flag.*
7. Target variable *'attack_cat'* is encoded using label encoder into 5 categories.
8. Final data set is prepared as *train* and *test*

Step2 . Feature selection using variable reduction yechniques

1. Recursive feature elimination

2. Seleck K best features

3. Variance Inflation factor analysis.

Recursive feature elimination and select k best features methods were used to select features and after combining these features VIF analysis is performed on selected variables to remove multicolinearity: Final 13features **selected** are.

- dst_host_same_src_port_rate
- dst_host_count
- dst_host_srv_diff_host_rate
- srv_count
- flag_RSTR
- src_bytes
- dst_host_srv_count
- diff_srv_rate
- srv_diff_host_rate
- dst_host_diff_srv_rate
- dst_host_rerror_rate

Step 3: MODEL BUILIDING (Random Forest, KNN, Naïve Bayes)

1. Scikit is used
2. Train Data is split into train and test and metrics are calculated.
3. models (RF, KNN) performance was checked with cross validation.

ROC_AUC SCORE TEST FILE

	RF	KNN
Dos 0	.87	.89
NORMAL 1	.75	.73
Prob 2	.66	.63
R2L 3	.50	.53
U2R 4	.5	.5

Comparative summary of Classification Report

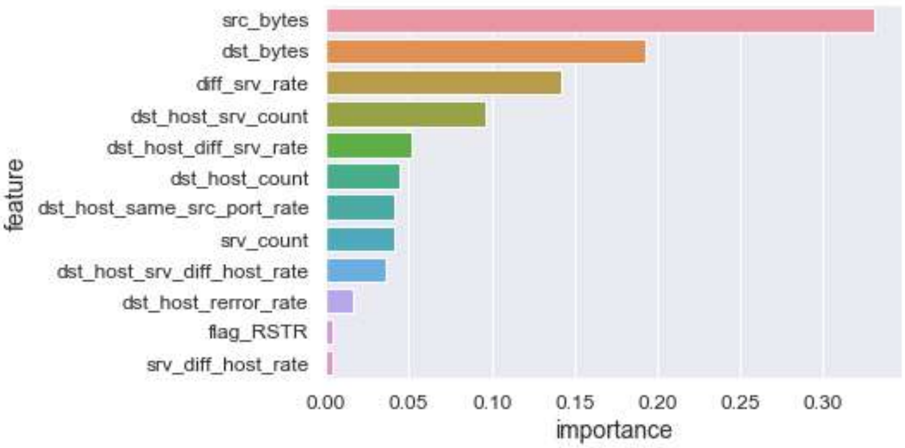
Test.txt File

	RF			KNN		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**

Dos 0	.91	.78	.84	.89	.83	.86
NORMAL 1	.62	.88	.73	.61	.94	.74
Prob 2	.80	.35	.49	.85	.28	.42
R2L 3	1	.04	.07	.94	.07	.13
U2R 4	0	0	0	0	0	0

RANDOM FOREST MODEL

Features with importance


src_bytes
dst_bytes
diff_srv_rate
dst_host_srv_count
dst_host_diff_srv_rate
dst_host_count
dst_host_same_src_port_rate
srv_count
dst_host_srv_diff_host_rate
dst_host_rerror_rate
flag_RSTR
srv_diff_host_rate
feature
importance