MACHINE LEARNING PROJET

Suprvised by :

Mrs. Trabelsi Wiem

esprit
Se former autrement

MEMBERS OF OUR GROUP

Chtourou Mohamed Jasser

Ben Rhaiem Ghofrane

Traidi Yassine

Ben Romdhane Aziz

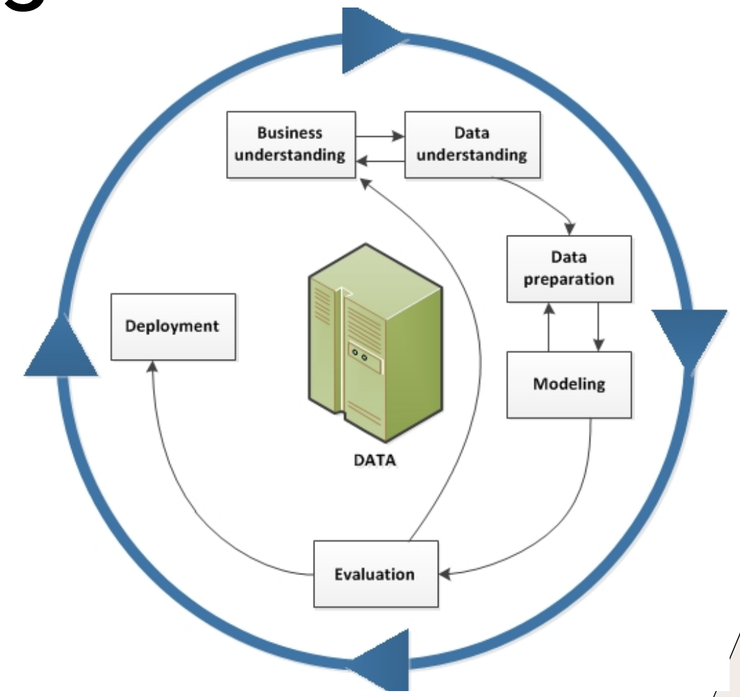
Jaouadi Yasmine

Gotrane Saifeddine

PRESENTATION PLAN

CRISP-DM STEPS

1- Business Understanding
2- Data Understanding
3- Data Preparation
4- Modeling
5- Evaluation
6- Deployment



LIBRARIES

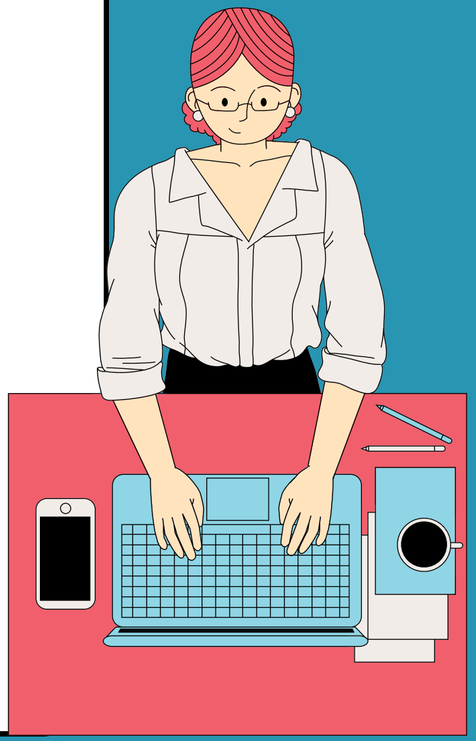
Seaborn

MatplotLib

Pandas

Sickit Learn

NumPy



BUSINESS UNDERSTANDING

- **Problem:**

Need to detect network intrusions and cyberattacks in real-time

- **Goals:**

-Understand network traffic behavior

-Identify anomalies and outliers indicating potential intrusions

-Improve security policies based on modeling results Success

- **Metrics:** Detection accuracy of intrusion types False positive/negative rates Time to detection of new intrusion patterns
- **Data access:** KDD Cup 99 dataset of network connections
- **Resources:** Stakeholder buy-in, computing resources

DATA UNDERSTANDING

What is the dimensions of our dataset?

```
train.shape
```

```
(125973, 43)
```

```
test.shape
```

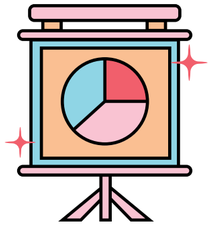
```
(22544, 43)
```

```
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   duration                     22544 non-null  int64
 1   protocol_type                22544 non-null  object
 2   service                      22544 non-null  object
 3   flag                         22544 non-null  object
 4   src_bytes                    22544 non-null  int64
 5   dst_bytes                    22544 non-null  int64
 6   land                         22544 non-null  int64
 7   wrong_fragment               22544 non-null  int64
 8   urgent                       22544 non-null  int64
 9   hot                          22544 non-null  int64
 10  num_failed_logins            22544 non-null  int64
 11  logged_in                    22544 non-null  int64
 12  num_compromised              22544 non-null  int64
 13  root_shell                   22544 non-null  int64
 14  su_attempted                 22544 non-null  int64
 15  num_root                     22544 non-null  int64
 16  num_file_creations           22544 non-null  int64
 17  num_shells                   22544 non-null  int64
 18  num_access_files             22544 non-null  int64
 19  num_outbound_cmds            22544 non-null  int64
 20  is_host_login                22544 non-null  int64
 21  is_guest_login               22544 non-null  int64
 22  count                        22544 non-null  int64
 23  srv_count                    22544 non-null  int64
 24  serror_rate                  22544 non-null  float64
 25  srv_serror_rate              22544 non-null  float64
 26  rerror_rate                  22544 non-null  float64
 27  srv_rerror_rate              22544 non-null  float64
 28  same_srv_rate                22544 non-null  float64
 29  diff_srv_rate                22544 non-null  float64
 30  srv_diff_host_rate           22544 non-null  float64
 31  dst_host_count               22544 non-null  int64
 32  dst_host_srv_count           22544 non-null  int64
 33  dst_host_same_srv_rate       22544 non-null  float64
 34  dst_host_diff_srv_rate       22544 non-null  float64
 35  dst_host_same_src_port_rate  22544 non-null  float64
 36  dst_host_srv_diff_host_rate  22544 non-null  float64
 37  dst_host_serror_rate         22544 non-null  float64
 38  dst_host_srv_serror_rate     22544 non-null  float64
 39  dst_host_rerror_rate         22544 non-null  float64
 40  dst_host_srv_rerror_rate     22544 non-null  float64
 41  attack                       22544 non-null  object
 42  level                        22544 non-null  int64
```

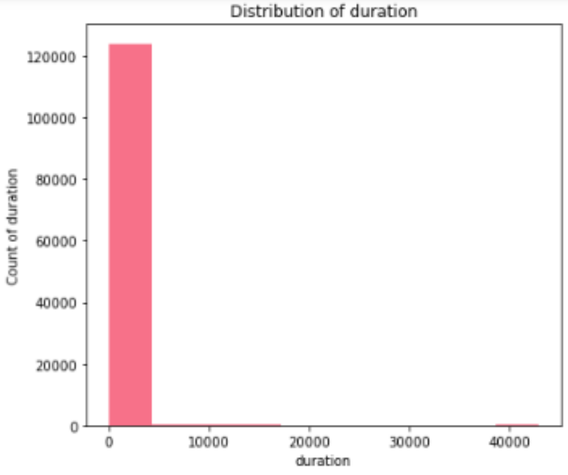
What are the distinct types of information available in our dataset?

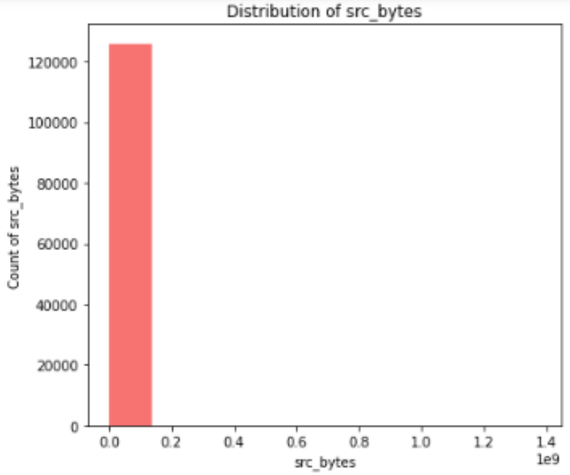
DATA UNDERSTANDING

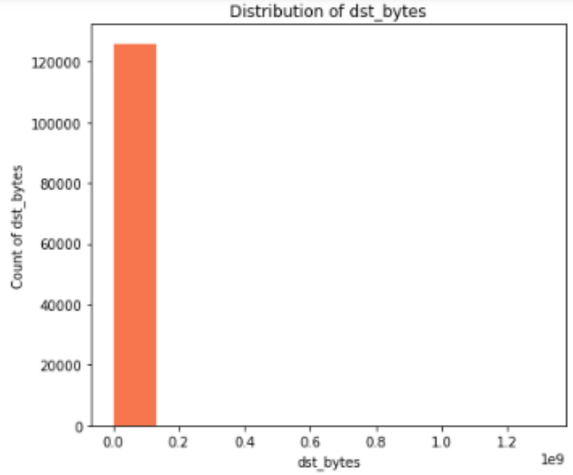
Data visualisation

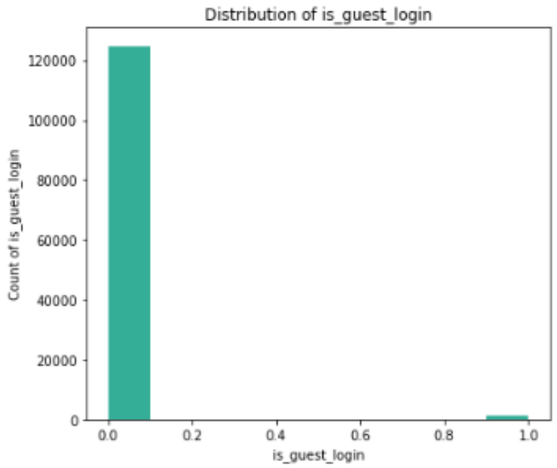


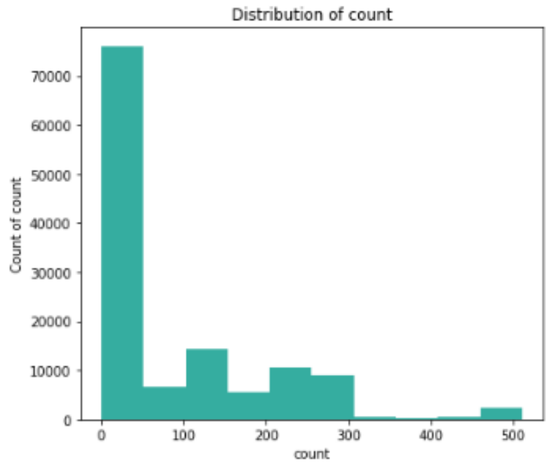
Some of Quantitative Variables

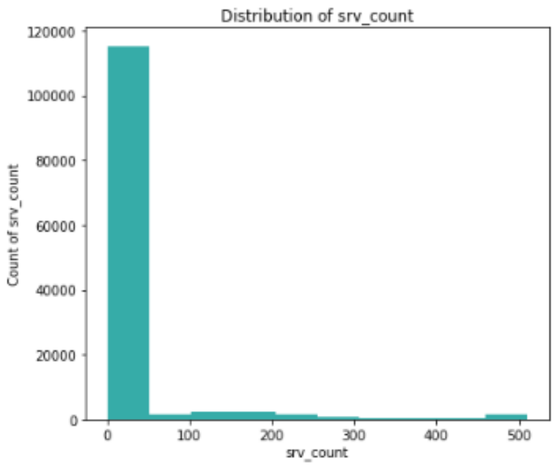








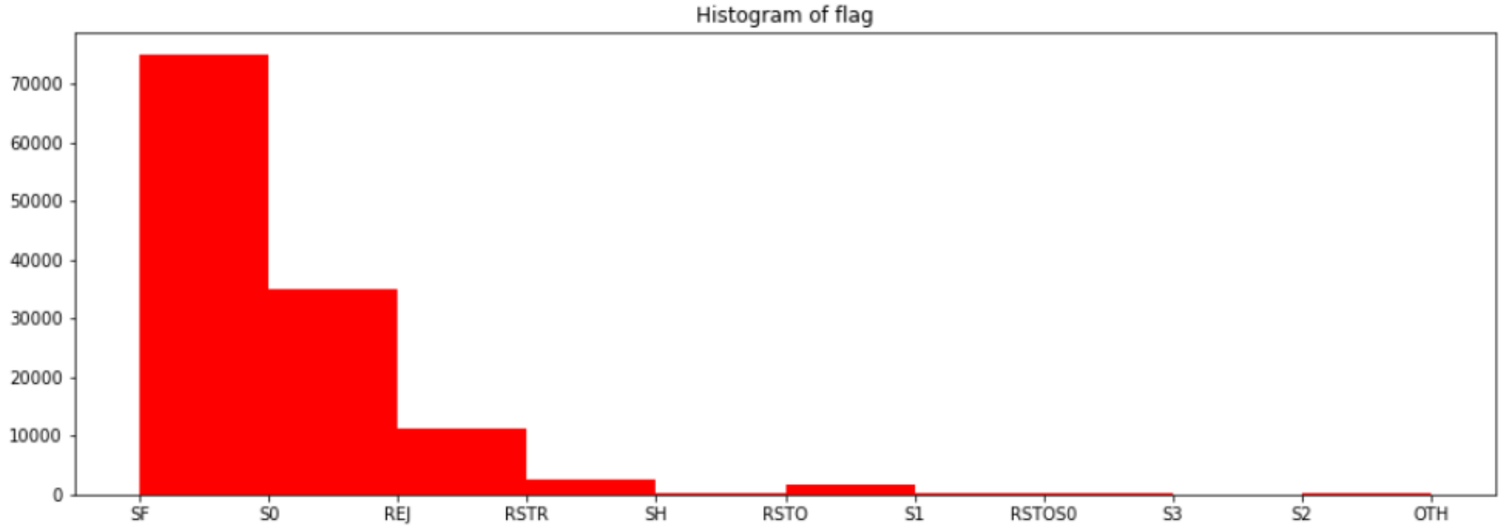


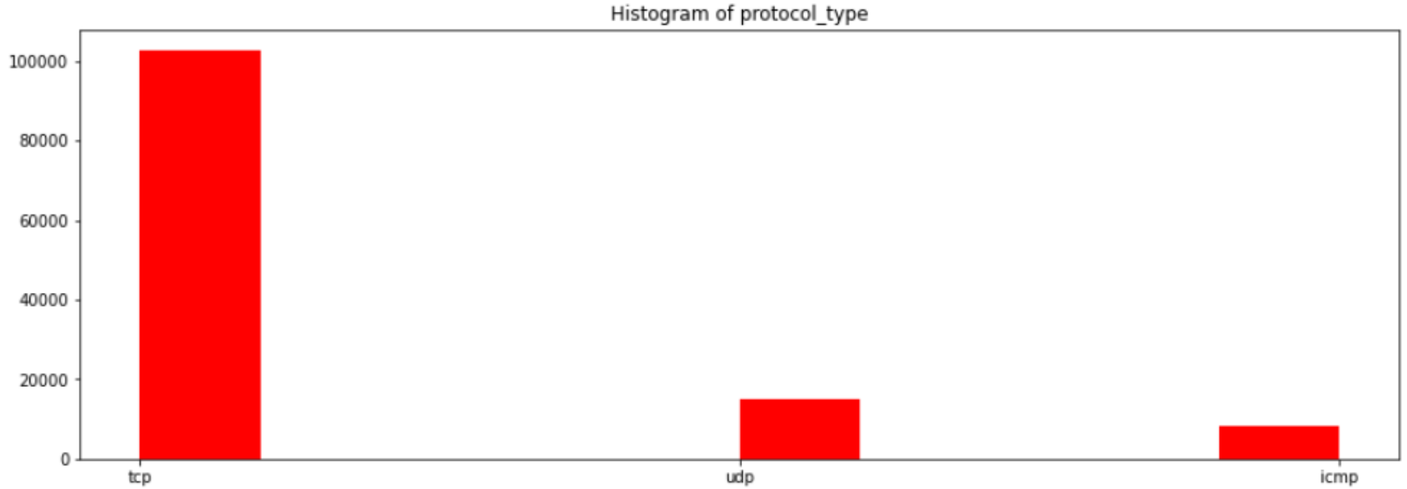


DATA UNDERSTANDING

Data visualisation

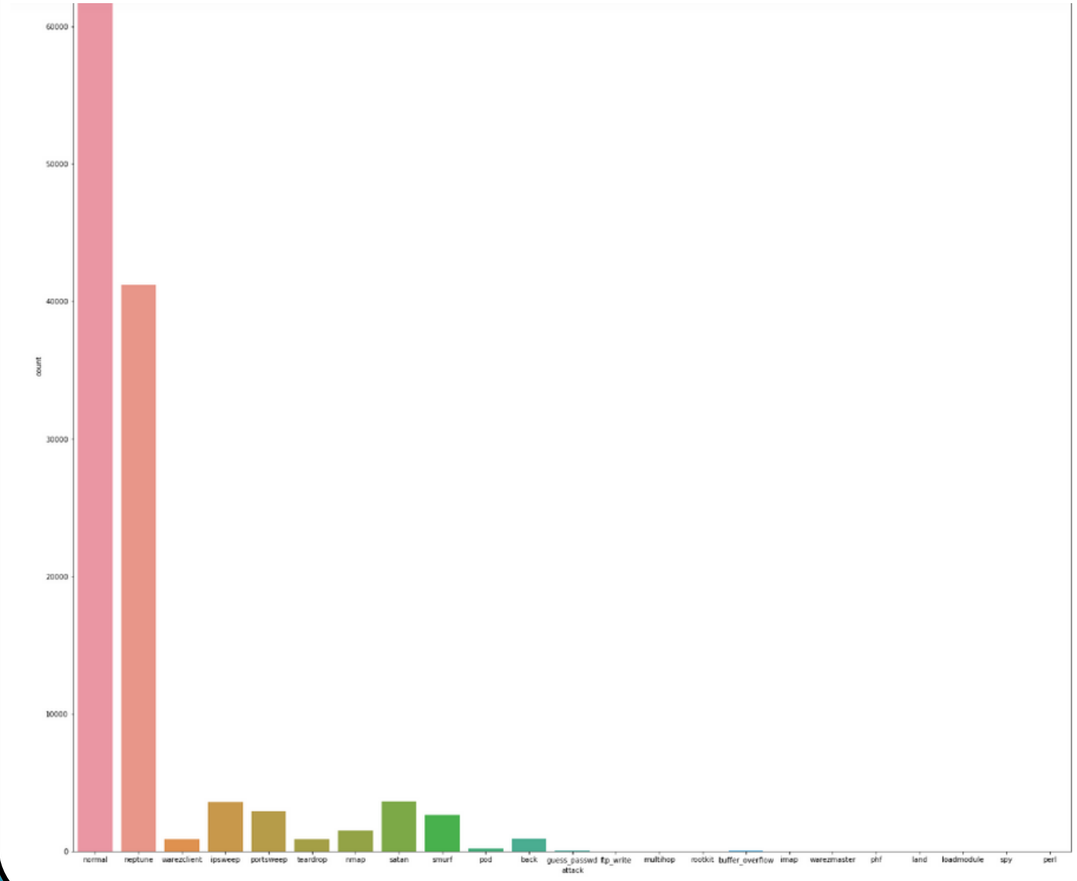
Some of Qualitative Variables



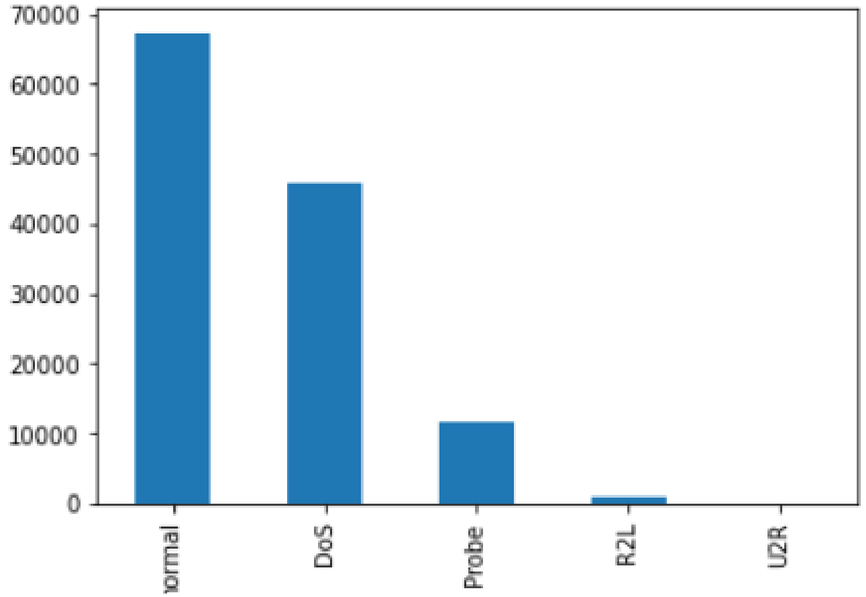


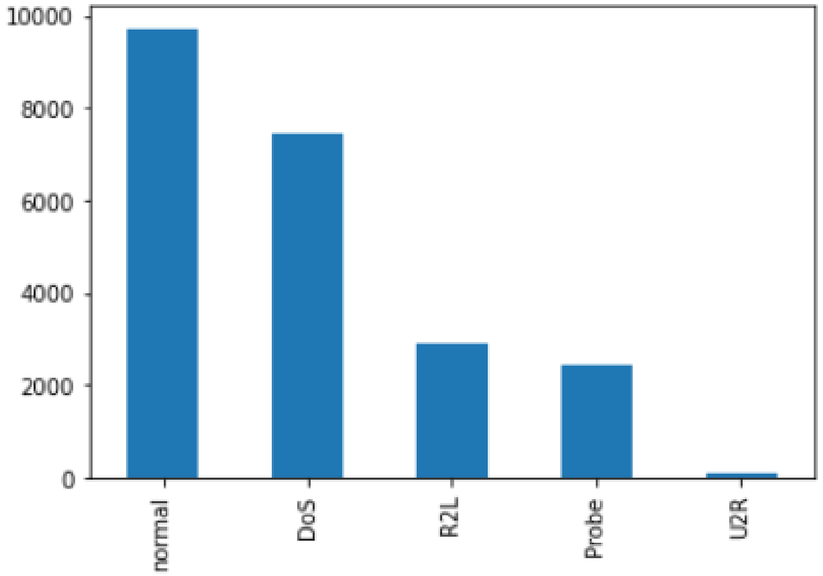
DATA UNDERSTANDING

Understanding the target variable : attack

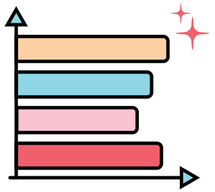


Train attack





Test attack



DATA PREPARATION

DATA CLEANING

Cheking for missing values :

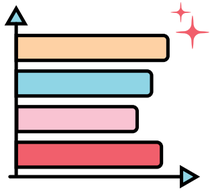
```
print(('We have {} missing values in out Train set\nWe have {} missing values in the Test set')
```

```
We have 0 missing values in out Train set
We have 0 missing values in the Test set
```

Cheking for duplicated values :

```
print(test.duplicated().sum())
print(train.duplicated().sum())
```

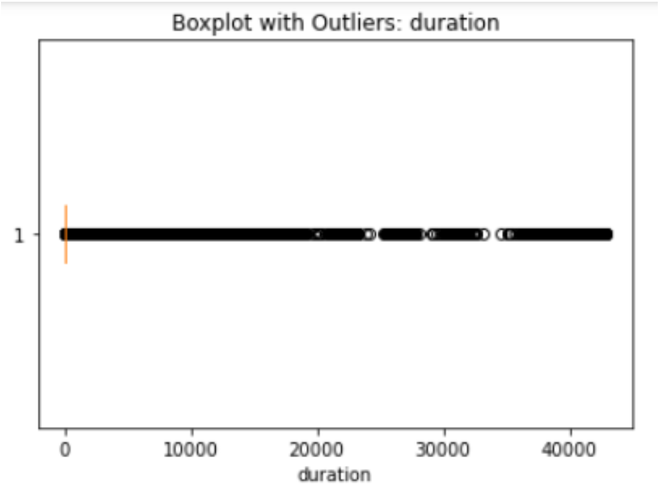
```
0
0
```

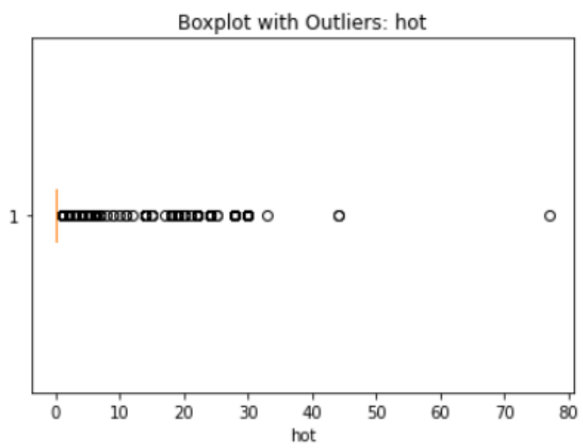


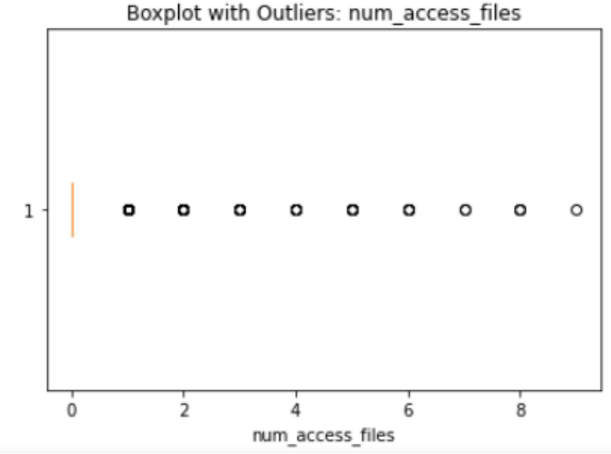
DATA PREPARATION

DATA CLEANING

Cheking for outliers :



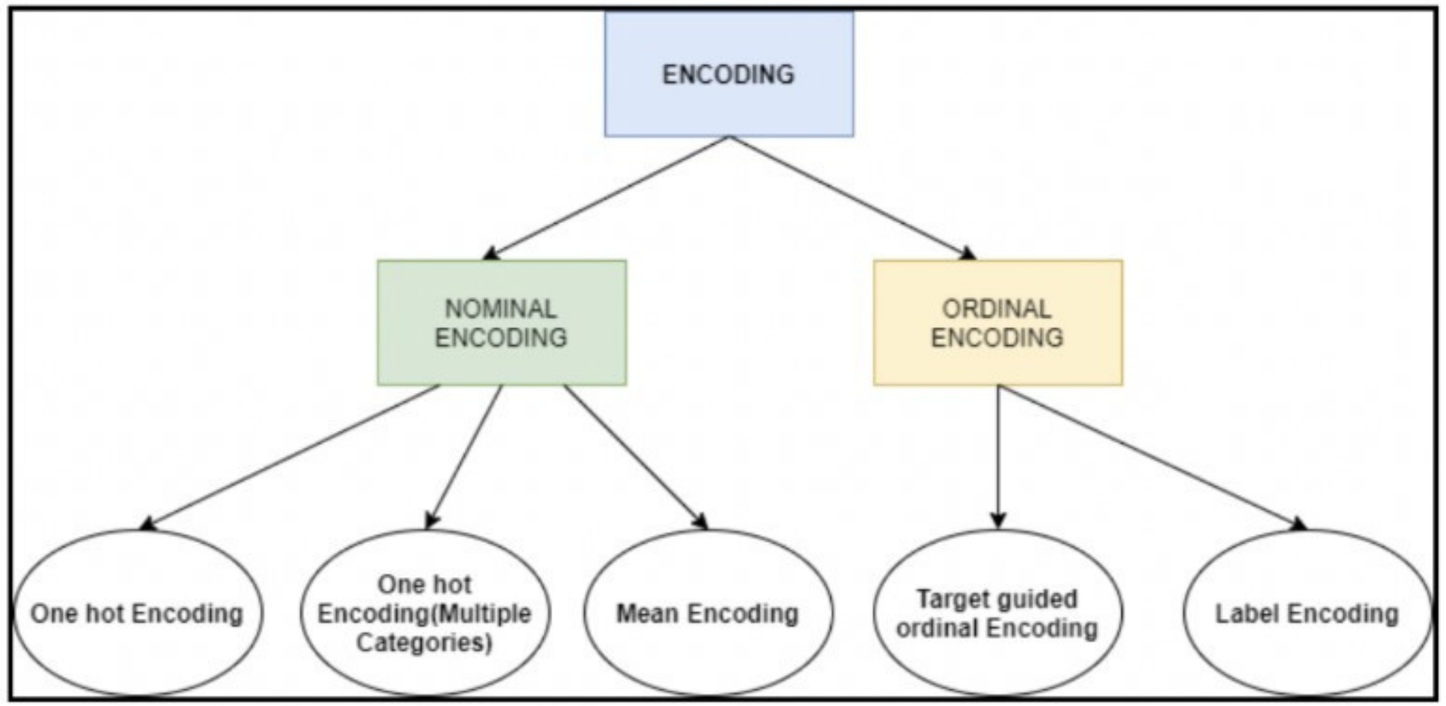




=> While our dataset contains outliers, we've chosen to retain them intentionally as they offer valuable insights or represent unique scenarios critical to understanding the complete spectrum of our data. Not to mention that the features are on the same scall so we don t need to scall it

DATA PREPARATION

DATA TRANSFORMATION



DATA PREPARATION

Nominal Features:

```
train_categorical_values.head()
```

	protocol_type	service	flag
0	tcp	ftp_data	SF
1	udp	other	SF
2	tcp	private	S0
3	tcp	http	SF
4	tcp	http	SF

Train

```
test_categorical_values.head()
```

	protocol_type	service	flag
0	tcp	private	REJ
1	tcp	private	REJ
2	tcp	ftp_data	SF
3	icmp	eco_i	SF
4	tcp	telnet	RSTO

Test

OneHotEncoding

	Protocol_type_icmp	Protocol_type_tcp	Protocol_type_udp	service_IRC	service_X11	service_Z39_50	service_aol	service_auth	service_bgp	service_courier
0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

5 rows × 84 columns

OneHotEncoding

	Protocol_type_icmp	Protocol_type_tcp	Protocol_type_udp	service_IRC	service_X11	service_Z39_50	service_auth	service_bgp	service_courier	service_csn
0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

5 rows × 78 columns

As we can see the test dataset has 78 features whereas train dataset has 84 so we need to add 6 features to balance the Service column

DATA PREPARATION

Nominal Features:

```
print(newtrain.shape)
print(newtest.shape)
```

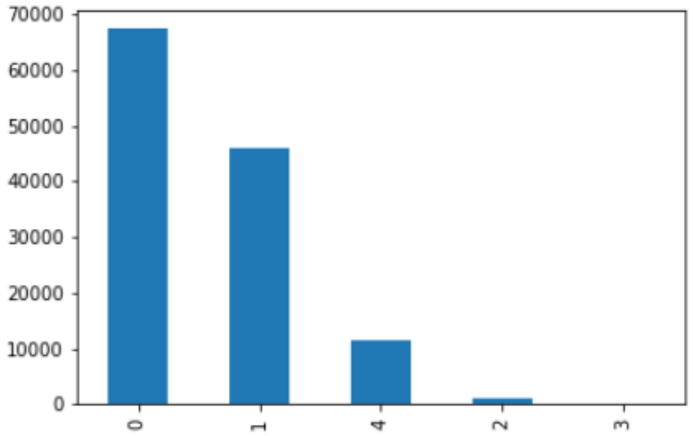
```
(125973, 122)
(22544, 122)
```

The test and train sets are balanced

	Total
normal	67343
DoS	45927
Probe	11656
R2L	995
U2R	52

Train attack

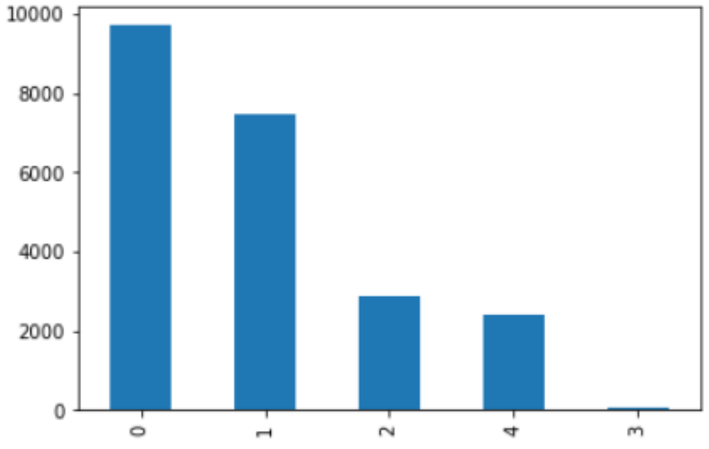
	Total
0	67343
1	45927
4	11656
2	995
3	52



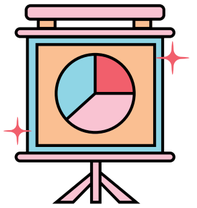
	Total
normal	9711
DoS	7460
R2L	2885
Probe	2421
U2R	67

Test attack

	Total
0	9711
1	7460
2	2885
4	2421
3	67



DATA PREPARATION



FEATURE SCALING

Splitting Dataset:

```
# Split dataframes into X & Y
X_DoS = DoS_df.drop('attack',1)
Y_DoS = DoS_df.attack
X_Probe = Probe_df.drop('attack',1)
Y_Probe = Probe_df.attack
X_R2L = R2L_df.drop('attack',1)
Y_R2L = R2L_df.attack
X_U2R = U2R_df.drop('attack',1)
Y_U2R = U2R_df.attack
# test set
X_DoS_test = DoS_df_test.drop('attack',1)
Y_DoS_test = DoS_df_test.attack
X_Probe_test = Probe_df_test.drop('attack',1)
Y_Probe_test = Probe_df_test.attack
X_R2L_test = R2L_df_test.drop('attack',1)
Y_R2L_test = R2L_df_test.attack
X_U2R_test = U2R_df_test.drop('attack',1)
Y_U2R_test = U2R_df_test.attack
```

This code separates features and outcome variables for various attack types (DoS, Probe, R2L, and U2R) in order to organize data for both training and testing datasets. Having distinct datasets for testing and training models is standard procedure in order to assess the models performance.

Scaling Dataset:

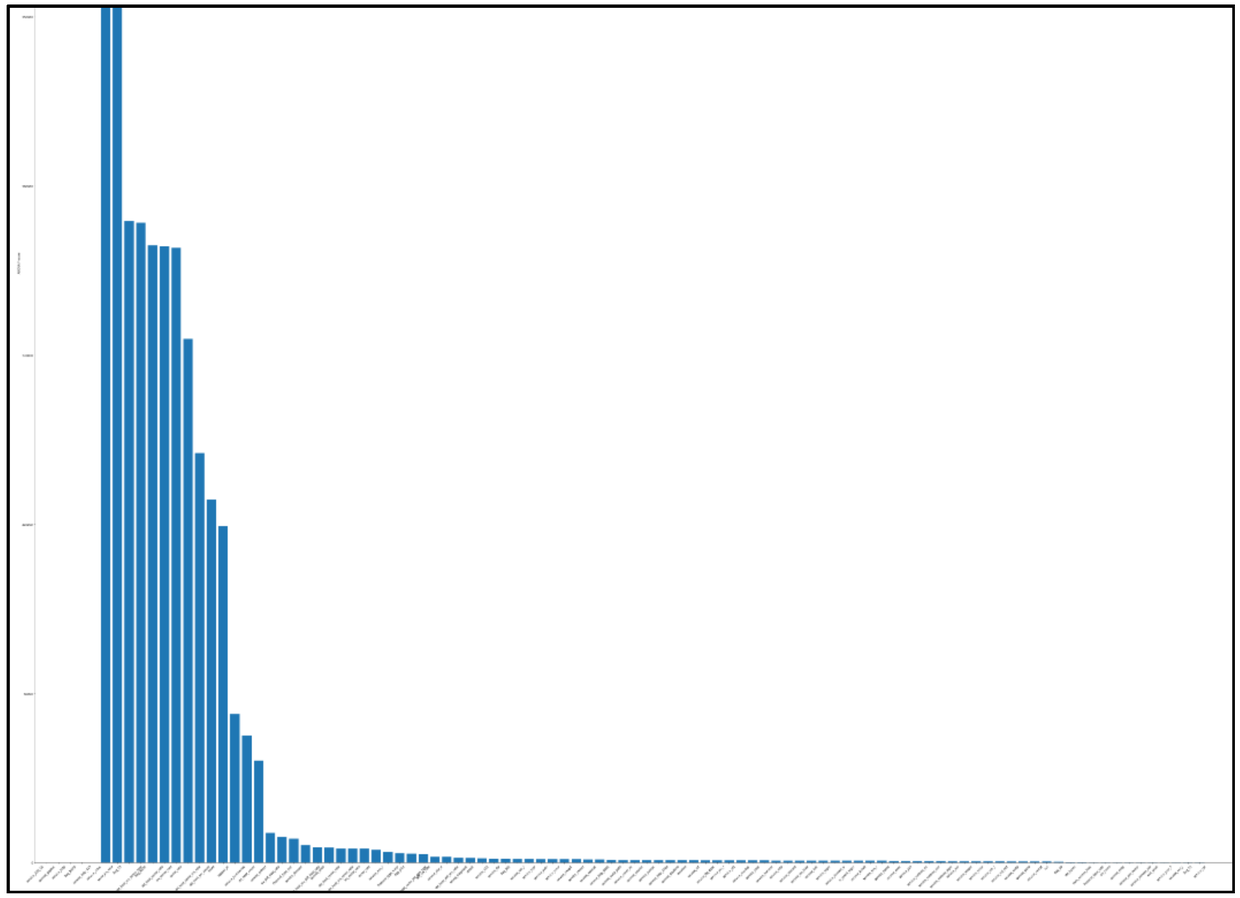
```
scaler1 = preprocessing.StandardScaler().fit(X_DoS)
X_DoS=scaler1.transform(X_DoS)
scaler2 = preprocessing.StandardScaler().fit(X_Probe)
X_Probe=scaler2.transform(X_Probe)
scaler3 = preprocessing.StandardScaler().fit(X_R2L)
X_R2L=scaler3.transform(X_R2L)
scaler4 = preprocessing.StandardScaler().fit(X_U2R)
X_U2R=scaler4.transform(X_U2R)
# test data
scaler5 = preprocessing.StandardScaler().fit(X_DoS_test)
X_DoS_test=scaler5.transform(X_DoS_test)
scaler6 = preprocessing.StandardScaler().fit(X_Probe_test)
X_Probe_test=scaler6.transform(X_Probe_test)
scaler7 = preprocessing.StandardScaler().fit(X_R2L_test)
X_R2L_test=scaler7.transform(X_R2L_test)
scaler8 = preprocessing.StandardScaler().fit(X_U2R_test)
X_U2R_test=scaler8.transform(X_U2R_test)
```

This code standardizes features in both training and test sets for each attack type. Standardization ensures features are on a similar scale, aiding the performance of models and algorithms.

DATA PREPARATION

FEATURE SELECTION

Feature selection primarily aims to choose the most important columns or features in a dataset to improve machine learning model performance, generalization, and efficiency, rather than for better readability or human understanding.



Transformation of the 'X_DoS' data to include only the selected features and shows the shape of the transformed data

```
X_newDoS = selector.fit_transform(X_DoS,Y_DoS)

print(X_newDoS.shape)

# Get the features that were selected: DoS

true=selector.get_support()
newcolindex_DoS=[i for i, x in enumerate(true) if x]
newcolname_DoS=list( colNames[i] for i in newcolindex_DoS )
```

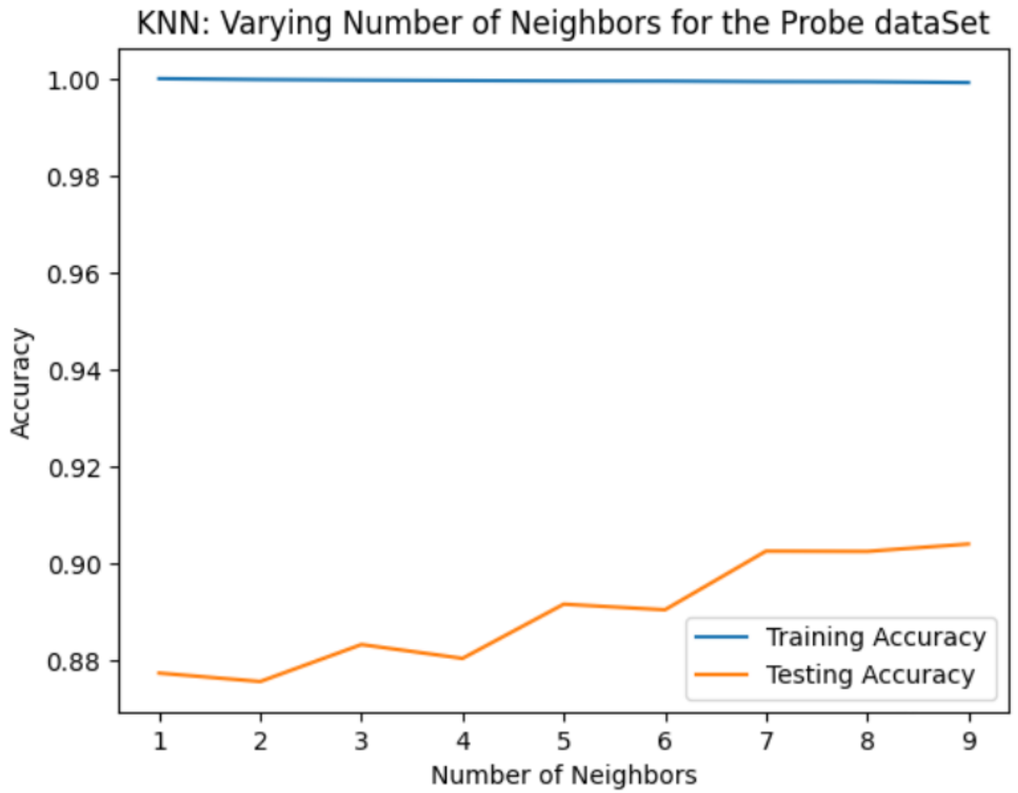
```
(113270, 12)
```

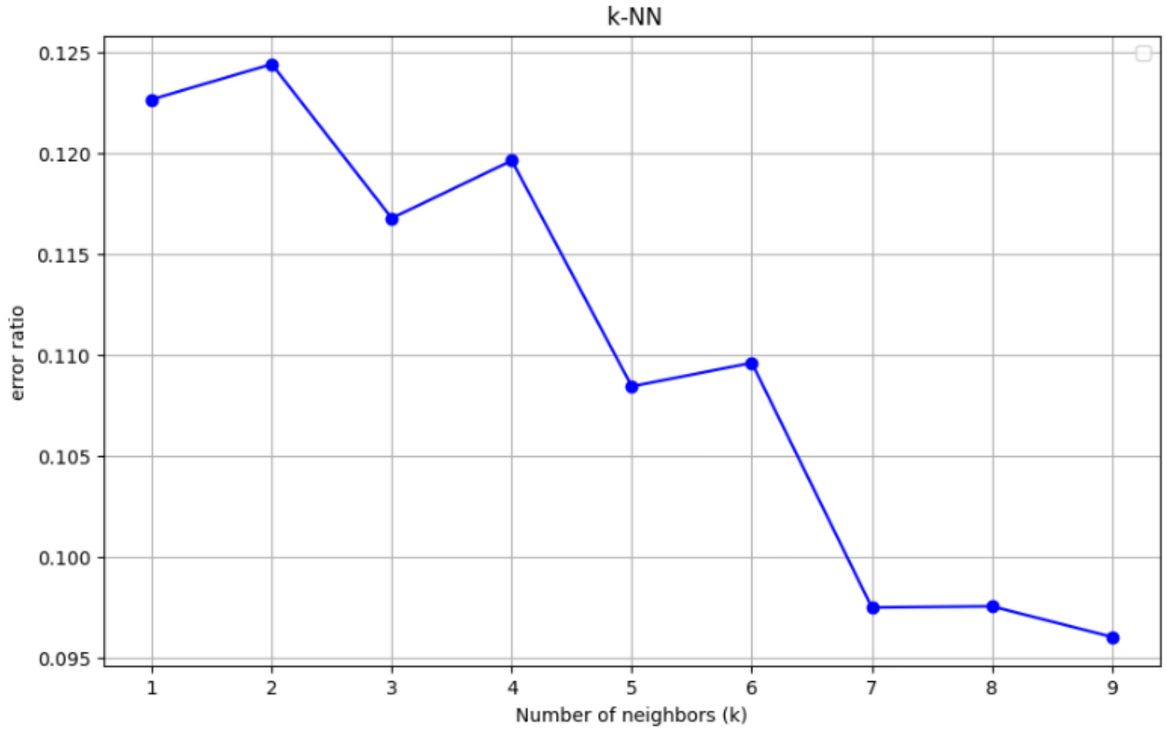
We applied the same process of feature selection to 'X_Probe,' 'X_R2L,' and 'X_U2R' as we did for 'X_DoS.'

MODELING

We're at the stage of selecting the initial modeling technique. We need to pinpoint the specific method, like deciding between using a decision-tree building or K-Nearest Neighbors.

So let’s start with the K-Nearest Neighbors





==> We can choose K= 7 or K=9 to have close results to the pdf but we opted for k = 5 to have exactly the same result as the pdf's ones. From the error ratio plot it's obvious that k=9 or k=7 have the smallest error rate but we have to choose 5 for identical results to the values mentioned in the pdf.

MODELING

CONFUSION MATRIX :

Is used in classification problems to assess where errors in the model were made, The confusion matrix presents the value of true positives, true negatives, false positives and false negatives.

Predicted attacks	0	1
Actual attacks		
0	9422	289
1	1573	5887

Dos

Accuracy: 0.99715
Precision: 0.99711
Recall: 0.99709
F-Score: 0.99710

Predicted attacks	0	4
Actual attacks		
0	9437	274
4	1272	1149

Probe

Accuracy: 0.99077
Precision: 0.98606
Recall: 0.98508
F-Score: 0.98553

Predicted attacks	0	2
Actual attacks		
0	9706	5
2	2883	2

R2L

Accuracy: 0.96713
Precision: 0.95271
Recall: 0.95456
F-Score: 0.95356

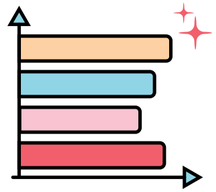
Predicted attacks	0	3
Actual attacks		
0	9711	0
3	65	2

U2R

Accuracy: 0.99703
Precision: 0.93143
Recall: 0.85073
F-Score: 0.87831

These are the result of the implementation of the algorithm KNN and the value of Accuracy, Precision, Recall and F-score for each kind of attack.

MODELING



Decision Tree:

Decision Trees classify by applying tests from the root to the leaves in a tree structure. Nodes test features, branches show outcomes, and leaves assign labels. They minimize uncertainty, often using entropy to measure disorder in the data

Dos

```
Accuracy: 0.99633
Precision: 0.99616
Recall: 0.99638
F-measure: 0.99627
```

Probe

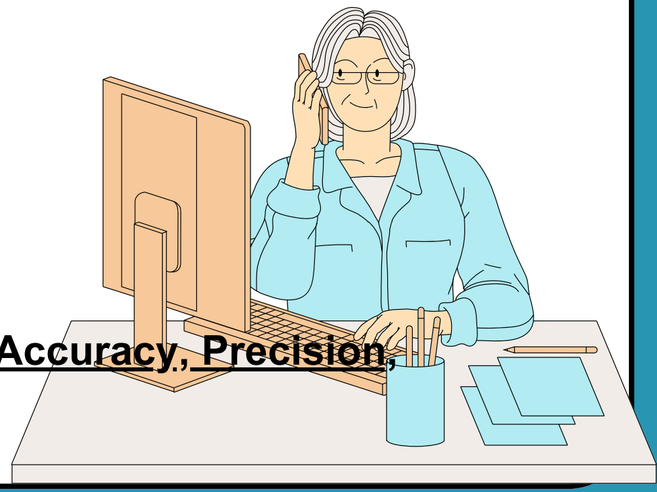
```
Accuracy: 0.99555
Precision: 0.99352
Recall: 0.99257
F-measure: 0.99303
```

R2L

```
Accuracy: 0.97952
Precision: 0.97196
Recall: 0.97002
F-measure: 0.97096
```

U2R

```
Accuracy: 0.99693
Precision: 0.88554
Recall: 0.91688
F-measure: 0.89612
```



These are the result of the implementation of the algorithm Decision Tree and the value of Accuracy, Precision, Recall and F-score for each kind of attack.

MODELING

CONFUSION MATRIX :

Predicted attacks	0	1
Actual attacks		
0	9497	214
1	2830	4630

Dos

Predicted attacks	0	4
Actual attacks		
0	2559	7152
4	186	2235

Probe

Predicted attacks	0	2
Actual attacks		
0	9704	7
2	2609	276

R2L

Predicted attacks	0	3
Actual attacks		
0	9703	8
3	48	19

U2R

MODELING

Support Vector Machine :

SVM aims to create the optimal decision boundary (hyperplane) in an n-dimensional space, effectively separating classes to correctly categorize new data points in the future.

Dos	Probe	R2L	U2R
Accuracy: 0.99371	Accuracy: 0.98450	Accuracy: 0.96793	Accuracy: 0.96793
Precision: 0.99342	Precision: 0.96907	Precision: 0.94854	Precision: 0.94854
Recall: 0.99380	Recall: 0.98365	Recall: 0.96264	Recall: 0.96264
F-measure: 0.99360	F-measure: 0.97613	F-measure: 0.95529	F-measure: 0.95529

CONFUSION MATRIX :

Predicted attacks / Actual attacks	0	1
0	9455	256
1	1359	6101

Predicted attacks / Actual attacks	0	4
0	9576	135
4	1285	1136

Predicted attacks / Actual attacks	0	2
0	9639	72
2	2737	148

Predicted attacks / Actual attacks	0	3
0	9710	1
3	67	0

MODELING

Random Forest:

Random Forest is a versatile machine learning algorithm used for both Classification and Regression tasks. It employs ensemble learning by combining multiple decision trees built on various subsets of the dataset. By averaging predictions from these trees, it improves predictive accuracy by considering the majority vote among the individual tree predictions

Dos	Probe	R2L	U2R
Accuracy: 0.99825	Accuracy: 0.99720	Accuracy: 0.98182	Accuracy: 0.99765
Precision: 0.99833	Precision: 0.99685	Precision: 0.97566	Precision: 0.95979
Recall: 0.99811	Recall: 0.99437	Recall: 0.97274	Recall: 0.86522
F-measure: 0.99822	F-measure: 0.99560	F-measure: 0.97418	F-measure: 0.90089

CONFUSION MATRIX :

Predicted attacks / Actual attacks	0	1
0	9680	31
1	6157	1303

Predicted attacks / Actual attacks	0	4
0	9408	303
4	994	1427

Predicted attacks / Actual attacks	0
0	9711
2	2885

Predicted attacks / Actual attacks	0
0	9711
3	67

MODELING

Logistic Regression:

Logistic regression, a widely used supervised learning algorithm, predicts categorical dependent variables based on a set of independent variables.

Dos

```
Accuracy: 0.99394
Precision: 0.99361
Recall: 0.99409
F-measure: 0.99384
```

Probe

```
Accuracy: 0.98442
Precision: 0.97105
Recall: 0.98096
F-measure: 0.97588
```

R2L

```
Accuracy: 0.96562
Precision: 0.94464
Recall: 0.96053
F-measure: 0.95221
```

U2R

```
Accuracy: 0.99683
Precision: 0.93066
Recall: 0.83763
F-measure: 0.86486
```

CONFUSION MATRIX :

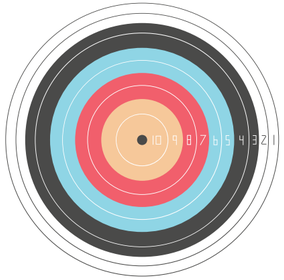
Predicted attacks	0	1
Actual attacks		
0	9680	31
1	6157	1303

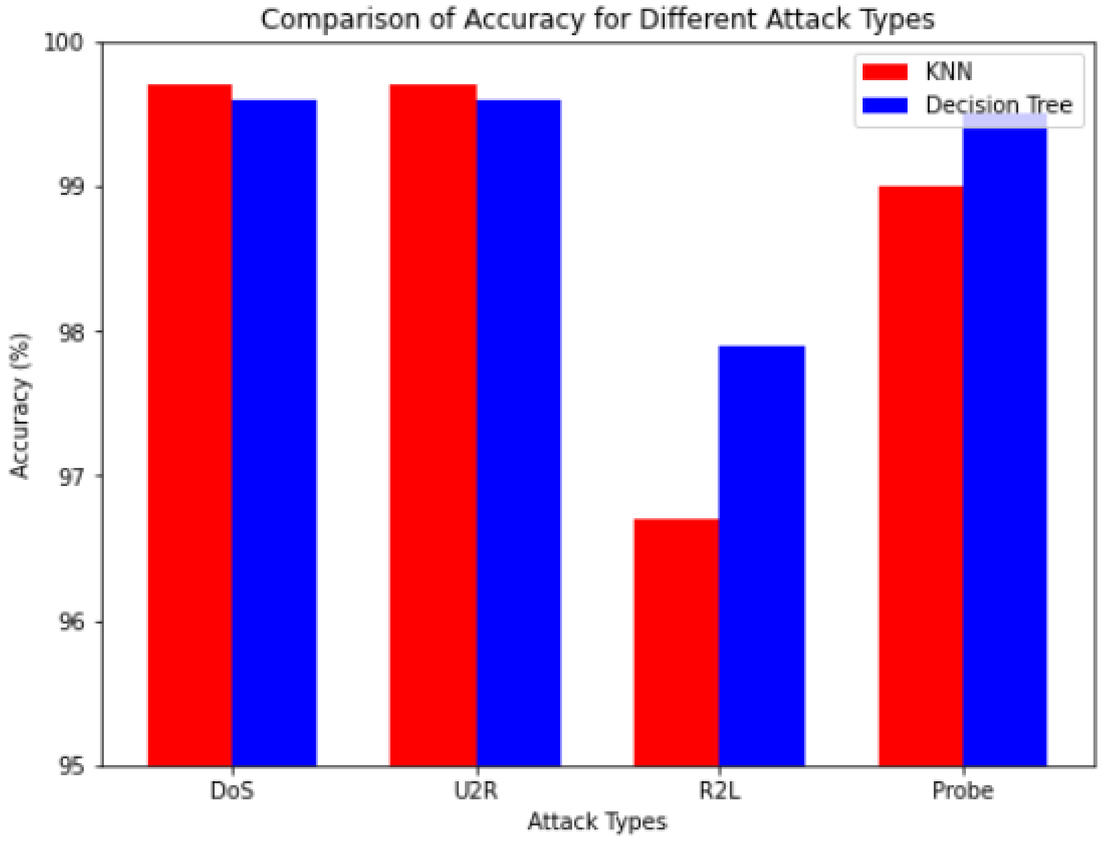
Predicted attacks	0	4
Actual attacks		
0	9562	149
4	1408	1013

Predicted attacks	0	2
Actual attacks		
0	9648	63
2	2764	121

Predicted attacks	0	3
Actual attacks		
0	9710	1
3	67	0

EVALUATION



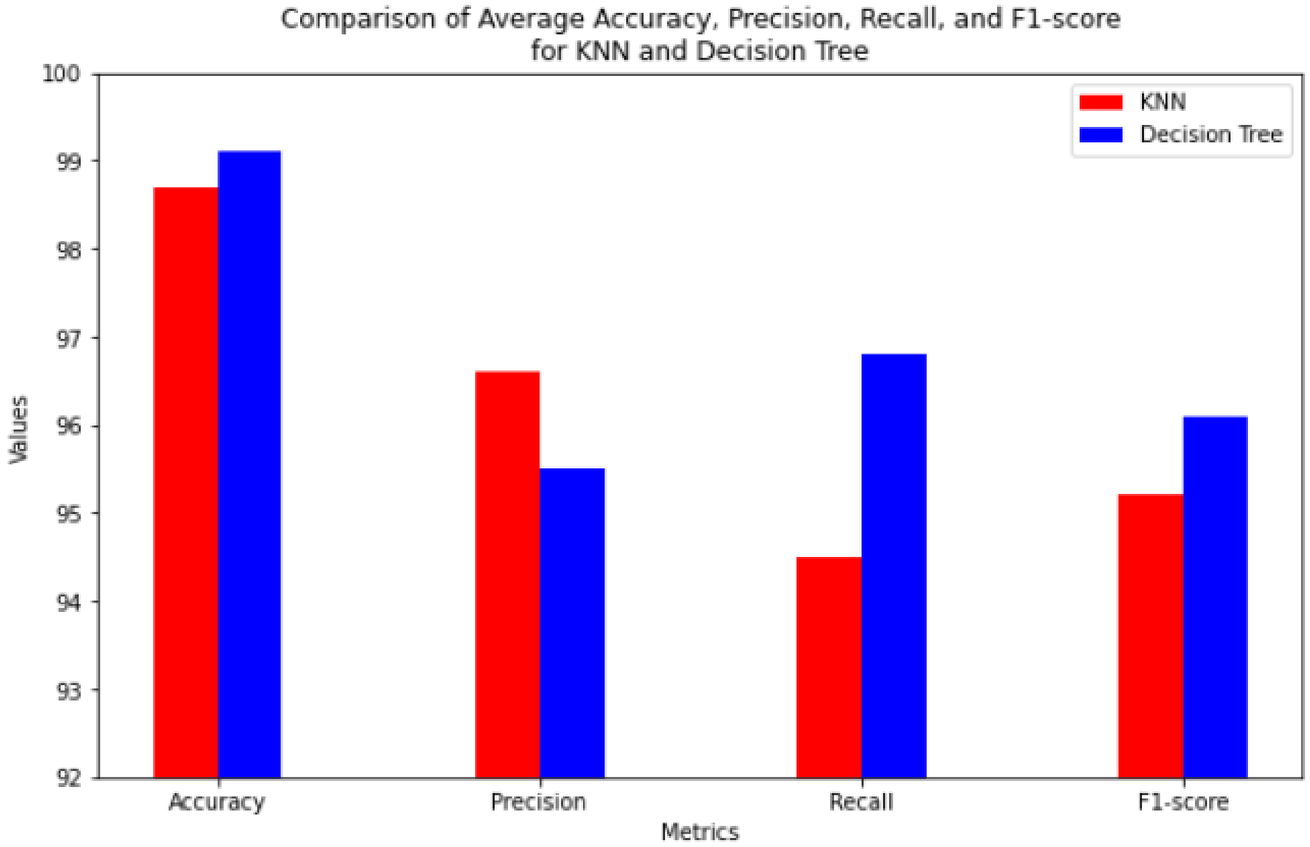


==> According to this plot KNN has slightly higher accuracy than decision tree in both DoS and U2R datasets which means that KNN is considered to be better than Decision tree at making correct predictions across all classes. But keep in mind that the difference is not so big.

Whereas in Probe and R2L datasets decision tree has a higher accuracy than KNN.It means that DT performed better than KNN in predicting correctly accross all classes in these 2 datasets.

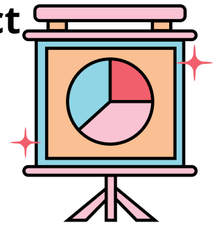
EVALUATION

Let's move to comparing the models by average accuracy which refers to a measure that considers the accuracy of a model for each class individually in a multi-class setting.

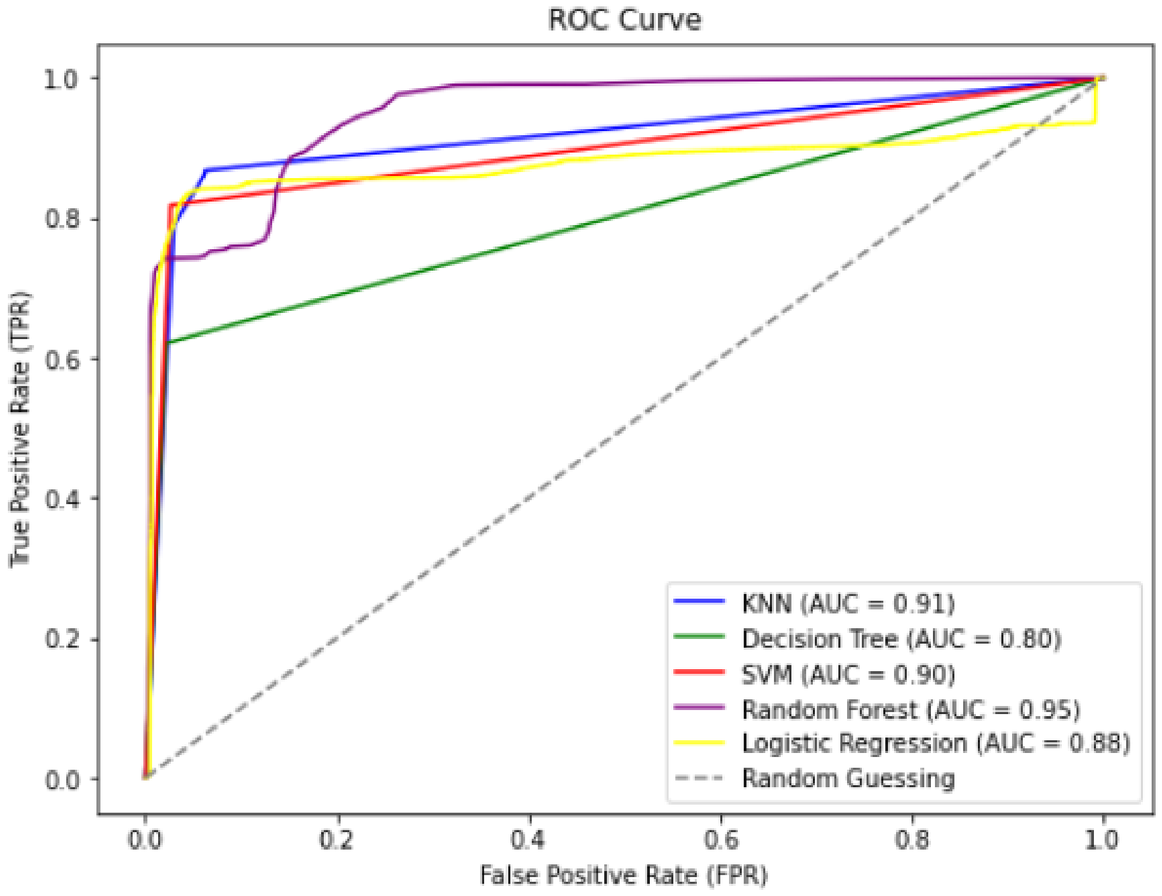


As we can see decision tree has a higher accuracy,recall and F1-score than KNN. But kNN has a higher precision than decision tree.

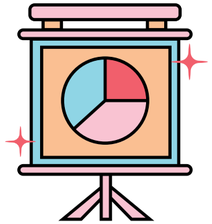
==> The Decision Tree outperforms kNN in terms of overall correctness (accuracy), ability to capture all instances of a particular class (recall), and the balanced trade-off between precision and recall (F1-score).
kNN excels in terms of precision, meaning that when it predicts positive instances, it is more likely to be correct than the Decision Tree



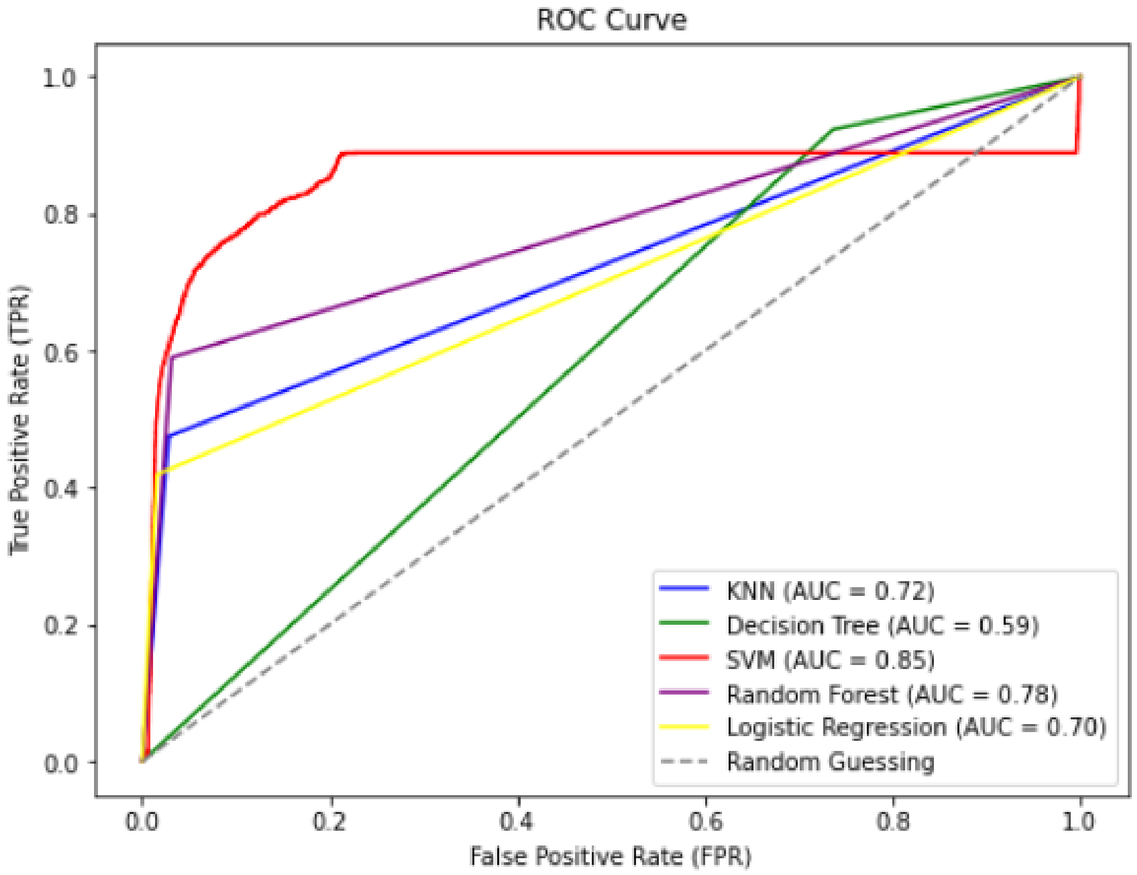
EVALUATION



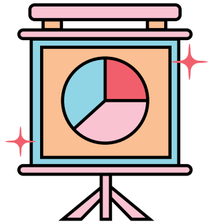
The ROC curves plot of DoS datasets shows that Random Forest has the largest area under curve which is equal to 0.95. It means that RF model has a better overall ability to distinguish between the positive and negative classes compared to the others.Then comes KNN with an AUC=0.91 and so on . The least is decision tree with an AUC equal to 0.8



EVALUATION

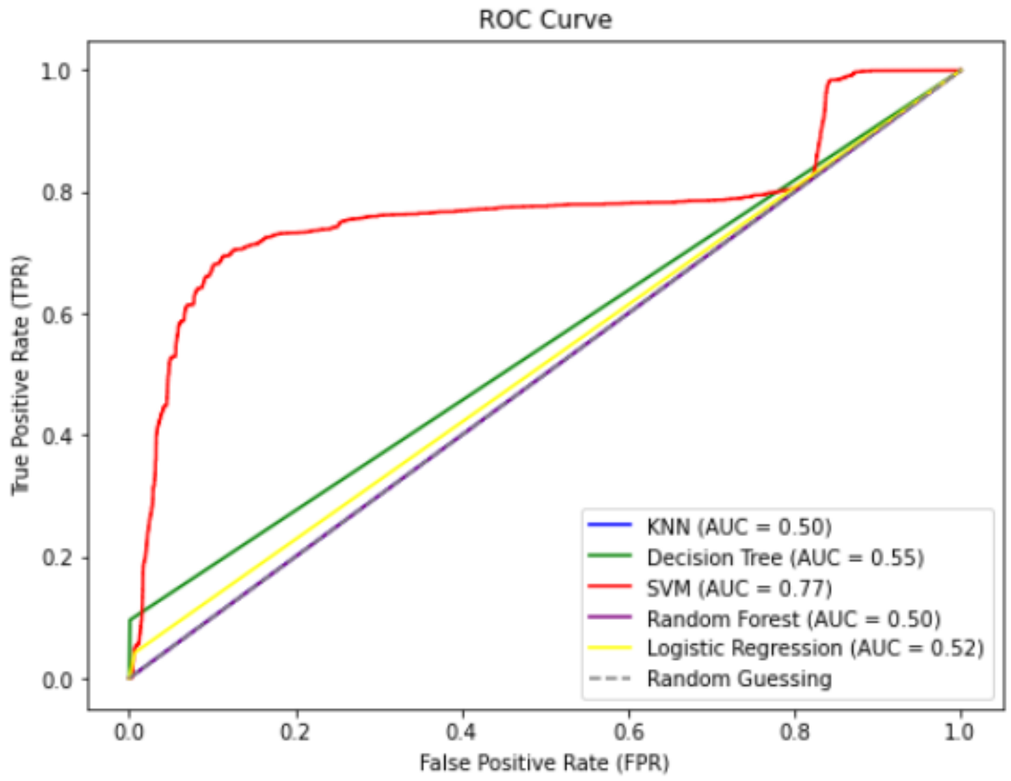


The ROC curves plot for Probe datasets shows that SVM has the largest area under curve which is equal to 0.95. It means that SVM model has a better overall ability to distinguish between the positive and negative classes compared to the others.Then comes RF with an AUC=0.78 and so on . The least is decision tree with an AUC equal to 0.59



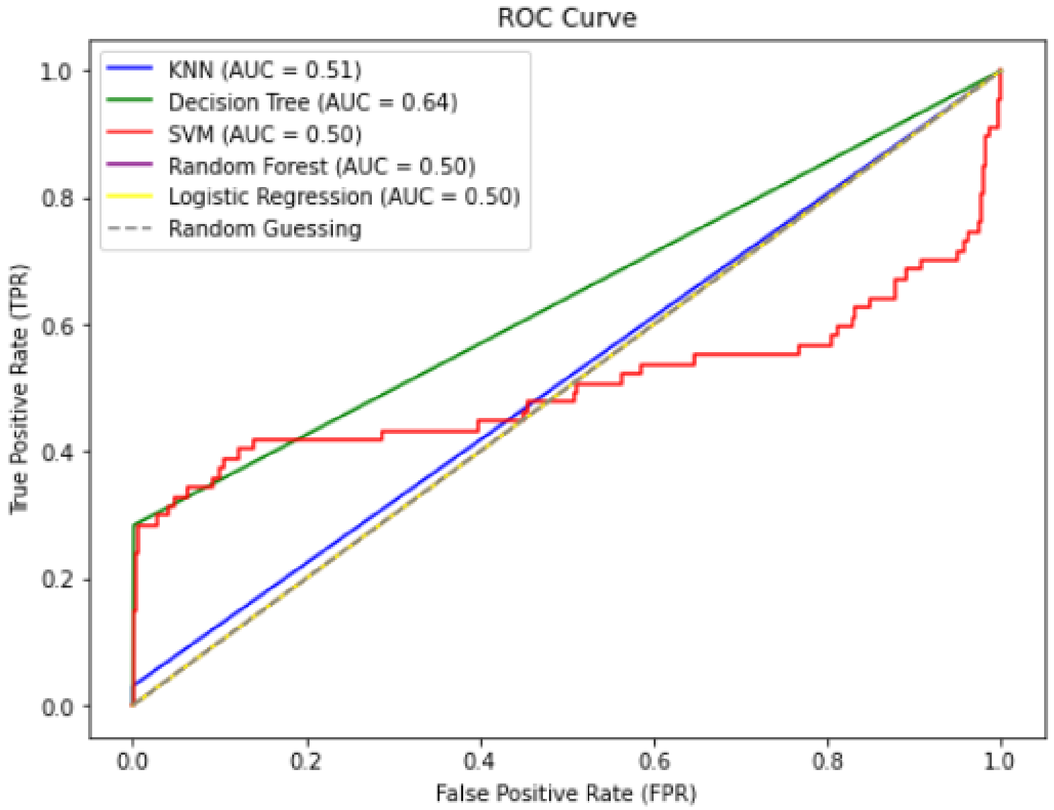
EVALUATION

The ROC curves plot for R2L datasets shows that SVM has the largest area under curve which is equal to 0.77. It means that SVM model has a better overall ability to distinguish between the positive and negative classes compared to the others.Then comes Decision tree with an AUC=0.55 and so on . The least is KNN and RF with an AUC equal to 0.5 . It means that KNN and RF's ability to distinguish between the positive and negative classes is no better than random guessing. Their ROC curve essentially hugs the diagonal line, and the model lacks discriminatory power

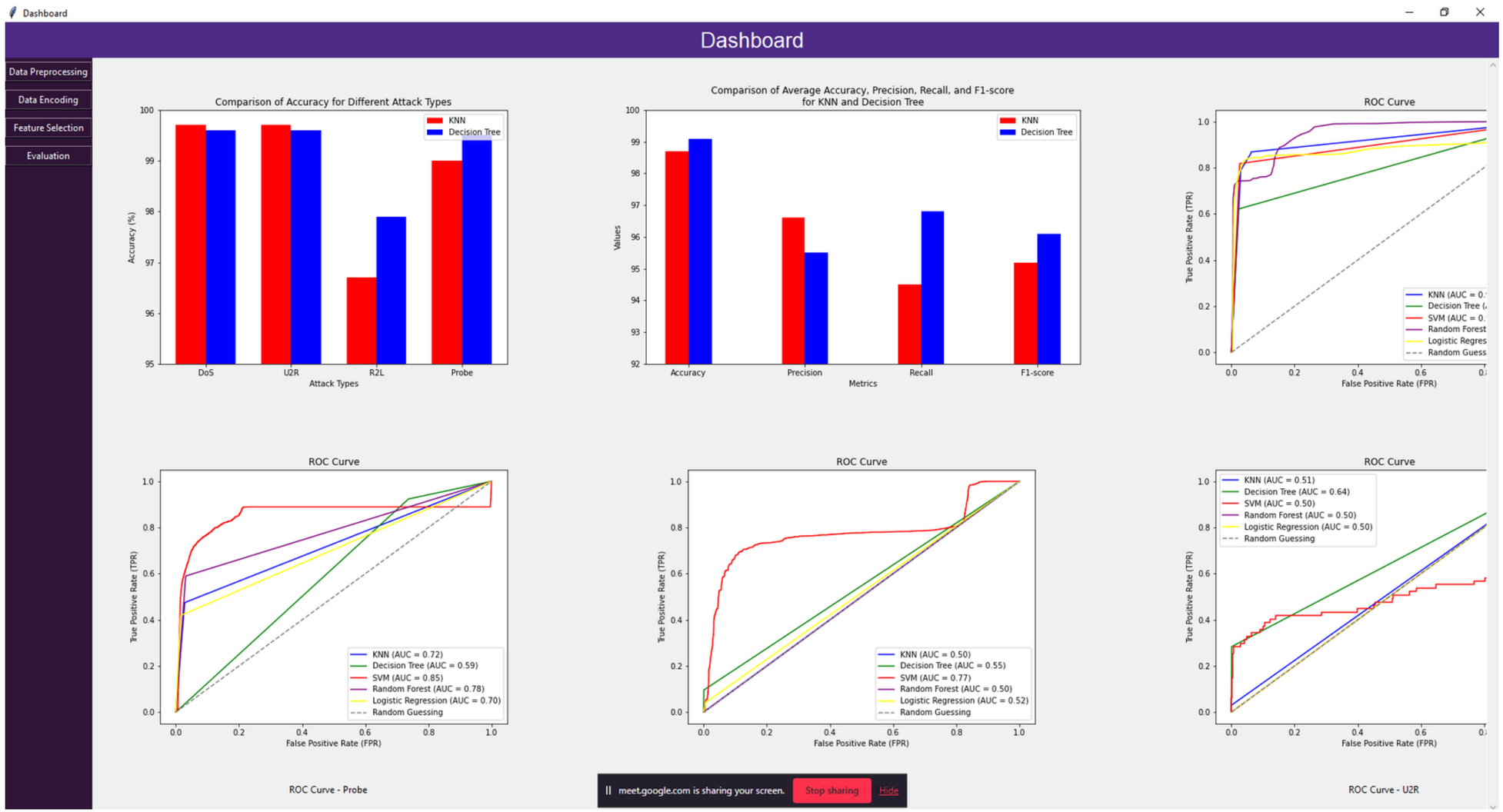


EVALUATION

The ROC curves plot for U2R datasets shows that decision tree has the largest area under curve which is equal to 0.64. It means that DT model has a better overall ability to distinguish between the positive and negative classes compared to the others.Then comes KNN with an AUC=0.51 and so on.



DEPLOYMENT


Dashboard
Dashboard
Data Preprocessing
Data Encoding
Feature Selection
Evaluation
Comparison of Accuracy for Different Attack Types
KNN
Decision Tree
Accuracy (%)
DoS
U2R
R2L
Probe
Attack Types
Comparison of Average Accuracy, Precision, Recall, and F1-score for KNN and Decision Tree
Values
Accuracy
Precision
Recall
F1-score
Metrics
ROC Curve
True Positive Rate (TPR)
False Positive Rate (FPR)
KNN (AUC = 0.72)
Decision Tree (AUC = 0.59)
SVM (AUC = 0.85)
Random Forest (AUC = 0.78)
Logistic Regression (AUC = 0.70)
Random Guessing
ROC Curve - Probe
KNN (AUC = 0.50)
Decision Tree (AUC = 0.55)
SVM (AUC = 0.77)
Random Forest (AUC = 0.50)
Logistic Regression (AUC = 0.52)
Random Guessing
KNN (AUC = 0.51)
Decision Tree (AUC = 0.64)
SVM (AUC = 0.50)
Random Forest (AUC = 0.50)
Logistic Regression (AUC = 0.50)
Random Guessing
ROC Curve - U2R
meet.google.com is sharing your screen.
Stop sharing
Hide

DEPLOYMENT


Network Attack Detector
Choose File
No file chosen
Predict
DoS Predictions: [1]
Probe Predictions: [4]
R2L Predictions: [0]
U2R Predictions: [0]