In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
path=r"C:\Users\gopun\Downloads\project 13\isd_project.csv"
dataset=pd.read_csv(path)
```

In [3]:
```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26013 entries, 0 to 26012
Data columns (total 42 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   duration                     26013 non-null  int64  
 1   protocol_type                26013 non-null  object 
 2   service                      26013 non-null  object 
 3   flag                         26013 non-null  object 
 4   src_bytes                    26013 non-null  int64  
 5   dst_bytes                    26013 non-null  int64  
 6   land                         26013 non-null  int64  
 7   wrong_fragment               26013 non-null  int64  
 8   urgent                       26013 non-null  int64  
 9   hot                          26013 non-null  int64  
 10  num_failed_logins            26013 non-null  int64  
 11  logged_in                    26013 non-null  int64  
 12  lnum_compromised             26013 non-null  int64  
 13  lroot_shell                  26013 non-null  int64  
 14  lsu_attempted                26013 non-null  int64  
 15  lnum_root                    26013 non-null  int64  
 16  lnum_file_creations          26013 non-null  int64  
 17  lnum_shells                  26013 non-null  int64  
 18  lnum_access_files            26013 non-null  int64  
 19  lnum_outbound_cmds           26013 non-null  int64  
 20  is_host_login                26013 non-null  int64  
 21  is_guest_login               26013 non-null  int64  
 22  count                        26013 non-null  int64  
 23  srv_count                    26013 non-null  int64  
 24  serror_rate                  26013 non-null  float64
 25  srv_serror_rate              26013 non-null  float64
 26  rerror_rate                  26013 non-null  float64
 27  srv_rerror_rate              26013 non-null  float64
 28  same_srv_rate                26013 non-null  float64
 29  diff_srv_rate                26013 non-null  float64
 30  srv_diff_host_rate           26013 non-null  float64
 31  dst_host_count               26013 non-null  int64  
 32  dst_host_srv_count           26013 non-null  int64  
 33  dst_host_same_srv_rate       26013 non-null  float64
 34  dst_host_diff_srv_rate       26013 non-null  float64
 35  dst_host_same_src_port_rate  26013 non-null  float64
 36  dst_host_srv_diff_host_rate  26013 non-null  float64
 37  dst_host_serror_rate         26013 non-null  float64
 38  dst_host_srv_serror_rate     26013 non-null  float64
 39  dst_host_rerror_rate         26013 non-null  float64
 40  dst_host_srv_rerror_rate     26013 non-null  float64
 41  label                        26013 non-null  object 
dtypes: float64(15), int64(23), object(4)
memory usage: 8.3+ MB
```

In [4]:
```
dataset.drop(['protocol_type','service','land','urgent',],axis=1,inplace=True)
```

In [5]:
```
dataset.drop(['hot','lnum_file_creations'],axis=1,inplace=True)
```

In [6]:
```
temp=pd.get_dummies(dataset['flag'],drop_first=True)
dataset=pd.concat([temp,dataset],axis=1)
dataset.drop('flag',axis=1,inplace=True)
```

In [7]:
```
x=dataset.iloc[:,0:44]
y=dataset.iloc[:,44]
```

In [8]:
```
x.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26013 entries, 0 to 26012
Data columns (total 44 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   REJ                          26013 non-null  uint8  
 1   RSTO                         26013 non-null  uint8  
 2   RSTOS0                       26013 non-null  uint8  
 3   RSTR                         26013 non-null  uint8  
 4   S0                           26013 non-null  uint8  
 5   S1                           26013 non-null  uint8  
 6   S2                           26013 non-null  uint8  
 7   S3                           26013 non-null  uint8  
 8   SF                           26013 non-null  uint8  
 9   SH                           26013 non-null  uint8  
 10  duration                     26013 non-null  int64  
 11  src_bytes                    26013 non-null  int64  
 12  dst_bytes                    26013 non-null  int64  
 13  wrong_fragment               26013 non-null  int64  
 14  num_failed_logins            26013 non-null  int64  
 15  logged_in                    26013 non-null  int64  
 16  lnum_compromised             26013 non-null  int64  
 17  lroot_shell                  26013 non-null  int64  
 18  lsu_attempted                26013 non-null  int64  
 19  lnum_root                    26013 non-null  int64  
 20  lnum_shells                  26013 non-null  int64  
 21  lnum_access_files            26013 non-null  int64  
 22  lnum_outbound_cmds           26013 non-null  int64  
 23  is_host_login                26013 non-null  int64  
 24  is_guest_login               26013 non-null  int64  
 25  count                        26013 non-null  int64  
 26  srv_count                    26013 non-null  int64  
 27  serror_rate                  26013 non-null  float64
 28  srv_serror_rate              26013 non-null  float64
 29  rerror_rate                  26013 non-null  float64
 30  srv_rerror_rate              26013 non-null  float64
 31  same_srv_rate                26013 non-null  float64
 32  diff_srv_rate                26013 non-null  float64
 33  srv_diff_host_rate           26013 non-null  float64
 34  dst_host_count               26013 non-null  int64  
 35  dst_host_srv_count           26013 non-null  int64  
 36  dst_host_same_srv_rate       26013 non-null  float64
 37  dst_host_diff_srv_rate       26013 non-null  float64
 38  dst_host_same_src_port_rate  26013 non-null  float64
 39  dst_host_srv_diff_host_rate  26013 non-null  float64
 40  dst_host_serror_rate         26013 non-null  float64
 41  dst_host_srv_serror_rate     26013 non-null  float64
 42  dst_host_rerror_rate         26013 non-null  float64
 43  dst_host_srv_rerror_rate     26013 non-null  float64
dtypes: float64(15), int64(19), uint8(10)
memory usage: 7.0 MB
```

In [9]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=
1)
```

In [10]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
x_train=sc.fit_transform(x_train)
x_test=sc.transform(x_test)
```

In [11]:
```
from sklearn.linear_model import LogisticRegression

reg=LogisticRegression()
reg.fit(x_train,y_train)
pred_logistic_reg=reg.predict(x_test)
```

```
F:\Anaconda\lib\site-packages\sklearn\linear_model\_logistic.py:940: Converge
nceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regres
sion
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
```

In [14]:
```
from sklearn.metrics import confusion_matrix,accuracy_score
score_logistic_reg=accuracy_score(y_test,pred_logistic_reg)*100
```

In [15]:
```
score_logistic_reg
```

Out[15]:
```
99.34648898001025
```

In [16]:
```
from sklearn.tree import DecisionTreeClassifier
classifier=DecisionTreeClassifier(criterion='entropy',random_state=0)
classifier.fit(x_train,y_train)
```

Out[16]:
```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='entropy',
                       max_depth=None, max_features=None, max_leaf_nodes=Non
e,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=0, splitter='best')
```

In [19]:
```
y_pred_decison_tree=classifier.predict(x_test)
score_decison_tree=accuracy_score(y_test,y_pred_decison_tree)*100
```

In [20]:
```
score_decison_tree
```

Out[20]:
```
99.6668375192209
```

In [21]:
```
from sklearn.ensemble import RandomForestClassifier
rfclassifier=RandomForestClassifier(n_estimators=10,criterion='entropy',random
_state=0)
rfclassifier.fit(x_train,y_train)
```

Out[21]:
```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='entropy', max_depth=None, max_features='aut
o',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=10,
                       n_jobs=None, oob_score=False, random_state=0, verbose=
0,
                       warm_start=False)
```

In [23]:
```
y_pred_random_forest=rfclassifier.predict(x_test)
score_random_forest=accuracy_score(y_test,y_pred_random_forest)*100
```

In [24]:
```
score_random_forest
```

Out[24]:
```
99.75653511019989
```

In [25]:
```
from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=5)
knn.fit(x_train,y_train)
```

Out[25]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                     weights='uniform')
```

In [26]:
```
y_pred_knn=knn.predict(x_test)
score_knn=accuracy_score(y_test,y_pred_knn)*100
```

In [27]:
```
score_knn
```

Out[27]:
```
99.39774474628396
```

In [28]:
```
from sklearn.naive_bayes import GaussianNB

model=GaussianNB()
model.fit(x_train,y_train)
```

Out[28]:
```
GaussianNB(priors=None, var_smoothing=1e-09)
```

In [29]:
```
y_pred_naive_bayes=model.predict(x_test)

score_naive_bayes=accuracy_score(y_test,y_pred_naive_bayes)*100
```

In [30]:
```
score_naive_bayes
```

Out[30]:
```
89.85135827780624
```