



```
import numpy as np
import pandas as pd
train = pd.read_csv('UNSW_NB15_training-set.csv')
test = pd.read_csv('UNSW_NB15_testing-set.csv')
from sklearn.preprocessing import OrdinalEncoder
ord_enc = OrdinalEncoder()
```

```
train['proto_code'] = ord_enc.fit_transform(train[['proto']])
train[['proto','proto_code']].head(175341)
train['state_code'] = ord_enc.fit_transform(train[['state']])
train[['state','state_code']].head(175341)
train['service_code'] = ord_enc.fit_transform(train[['service']])
train[['service','service_code']].head(175341)
```

	service	service_code	
0	-	0.0	
1	-	0.0	
2	-	0.0	
3	ftp	3.0	
4	-	0.0	
...	
175336	dns	2.0	
175337	-	0.0	
175338	dns	2.0	
175339	dns	2.0	
175340	dns	2.0	

175341 rows × 2 columns

```
test['proto_code'] = ord_enc.fit_transform(test[['proto']])
test[['proto','proto_code']].head(82332)
test['state_code'] = ord_enc.fit_transform(test[['state']])
test[['state','state_code']].head(82332)
test['service_code'] = ord_enc.fit_transform(test[['service']])
test[['service','service_code']].head(82332)
```

	service	service_code	
0	-	0.0	
1	-	0.0	
2	-	0.0	
3	-	0.0	
4	-	0.0	
...	
82327	-	0.0	
82328	-	0.0	
82329	-	0.0	
82330	-	0.0	
82331	-	0.0	

82332 rows × 2 columns

```
x1 = train[['dur','spkts','dpkts','proto_code','state_code','service_code',
            'sbytes','dbytes','rate','sttl','dttl','sload','dload','sloss','dloss',
            'sinpkt','dinpkt','sjit','djit','swin','stcpb',
            'dtcpb','dwin','tcprrt','synack','ackdat','smean','dmean','trans_depth','response_body_len','ct_srv_src','ct_state_t',
            'ct_dst_ltm','ct_src_dport_ltm','ct_dst_sport_ltm','ct_dst_src_ltm','is_ftp_login',
            'ct_ftp_cmd','ct_flw_http_mthd','ct_src_ltm','ct_srv_dst','is_sm_ips_ports']]
y1 = train['label']
x2 = test[['dur','spkts','dpkts','proto_code','state_code','service_code',
            'sbytes','dbytes','rate','sttl','dttl','sload','dload','sloss','dloss',
            'sinpkt','dinpkt','sjit','djit','swin','stcpb',
            'dtcpb','dwin','tcprrt','synack','ackdat','smean','dmean','trans_depth','response_body_len','ct_srv_src','ct_state_t',
            'ct_dst_ltm','ct_src_dport_ltm','ct_dst_sport_ltm','ct_dst_src_ltm','is_ftp_login',
```

```
        'ct_ftp_cmd','ct_flw_http_mthd','ct_src_ltm','ct_srv_dst','is_sm_ips_ports']]
y2 = test['label']

from sklearn.preprocessing import MinMaxScaler
model = MinMaxScaler()
model.fit(x1)
x1 = model.transform(x1)
x2 = model.transform(x2)
```

```
from sklearn.tree import DecisionTreeClassifier
ML=DecisionTreeClassifier()
```

```
ML=ML.fit(x1,y1)
```

```
y_pred=ML.predict(x2)
print("Class Predicted: ",y_pred)
```

Class Predicted: [0 0 1 ... 0 0 1]

```
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y2, y_pred)*100
print("Accuracy = ",accuracy)
from sklearn.metrics import f1_score
from sklearn import metrics
print("Confusion Matrix =", metrics.confusion_matrix(y2, y_pred, labels=None,
                                                    sample_weight=None))

print("Recall =", metrics.recall_score(y2, y_pred, labels=None,
                                      pos_label=1, average='weighted',
                                      sample_weight=None))

print("Precision =", metrics.precision_score(y2, y_pred, labels=None,
                                             pos_label=1, average='weighted',
                                             sample_weight=None))

print("Classification Report =\n", metrics.classification_report(y2, y_pred,
                                                                labels=None,
                                                                target_names=None,
                                                                sample_weight=None,
                                                                digits=2,
                                                                output_dict=False))

print("F1 Score = ",f1_score(y2, y_pred, average='macro'))
```

Accuracy = 85.6592819316912
Confusion Matrix = [[27233 9767]
 [2040 43292]]
Recall = 0.856592819316912
Precision = 0.8673284530741059
Classification Report =

	precision	recall	f1-score	support
0	0.93	0.74	0.82	37000
1	0.82	0.95	0.88	45332
accuracy			0.86	82332
macro avg	0.87	0.85	0.85	82332
weighted avg	0.87	0.86	0.85	82332

F1 Score = 0.8509210848655082

```
import graphviz
from sklearn import tree
data=tree.export_graphviz(ML,feature_names=['dur','spkts','dpkts','proto_code','state_code','service_code',
        'sbytes','dbytes','rate','sttl','dttl','sload','dload','sloss','dloss',
        'sinpkt','dinpkt','sjit','djit','swin','stcpb',
        'dtcpb','dwin','tcprrt','synack','ackdat','smean','dmean','trans_depth','response_body_len','ct_srv_src','ct_state',
        'ct_dst_ltm','ct_src_dport_ltm','ct_dst_sport_ltm','ct_dst_src_ltm','is_ftp_login',
        'ct_ftp_cmd','ct_flw_http_mthd','ct_src_ltm','ct_srv_dst','is_sm_ips_ports'],class_names=['0','1'],
        filled=True,rounded=False,special_characters=True)
op1=graphviz.Source(data)
op1
op1.render("Attack classification tree graph")
```

'Attack classification tree graph.pdf'

✓ 57s completed at 5:15 PM

