CSE-3024 Web Mining

Lab Assignment 8

Alokam Nikhitha

19BCE2555

Experiment 8

Aim

Using a Decision Tree Classifier, divide the given network intrusion dataset into normal and abnormal categories. Along with the classification, the following items must be printed:

- **Confusion Matrix**
- **Accuracy of model on Test data**
- **Decision Tree visualization.**

Dataset Used: The network intrusion dataset from Kaggle.

Link to which is:
https://www.kaggle.com/datasets/sampadab17/network-intrusion-detection?select=Train_data.csv

Procedure:

- First, we import the necessary numpy, pandas, matplotlib, and tree libraries.

- The dataset is then imported into our workspace. The set of independent and dependent attributes is also defined.

- Next, we used a 7.5:2.5 ratio to divide the dataset into training and test sets.

- Then, using DecisionTreeClassifier from sklearn.tree, we train our decision tree model.

- Next, we look for the test set results that our model anticipated.

- Then, using the expected and test set findings, we print our confusion matrix.

- Similarly, we print the model's accuracy based on the test set and anticipated result.

- Finally, we visualise our model using the sklearn tree.

Code:

```
#19BCE2555
#Importing libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import tree


#Importing dataset
dataset = pd.read_csv("Train_data.csv")
X = dataset.iloc[:, 4:41].values
y = dataset.iloc[:, -1].values


#Splitting the dataset
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
random_state=0)


#Fitting our model
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy' ,random_state = 0)
classifier.fit (X_train, y_train)


#Predicting the Test set Results
```

```
y_pred = classifier.predict(X_test)


#Printing the confusion matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)


#Printing the accuracy of our model
from sklearn.metrics import accuracy_score
accuracy = accuracy_score(y_test, y_pred)
print(accuracy)


#Defining the labels of our dataset
classes = ["Anamoly", "Normal"]


#Printing the visualized decision tree
fig = plt.figure(figsize=(25,20))
_ = tree.plot_tree(classifier,
               feature_names=dataset.columns,
               class_names=classes,
               filled=True)


#Printing the feature wise break points of our decision tree
test_representation = tree.export_text(classifier)
print(test_representation)
```

Code Snippets and Outputs:

```
In [1]: #19BCE2555
        #Importing libraries
        import numpy as np
        import matplotlib.pyplot as plt
        import pandas as pd
        from sklearn import tree
```

We're importing our libraries right now. Nupmy is imported as np, pandas is imported as pd, matplotlib's pyplot extension is imported as plt, and finally tree is imported from sklearn.

```
In [2]: #Importing dataset
        dataset = pd.read_csv("Train_data.csv")
        X = dataset.iloc[:, 4:41].values
        y = dataset.iloc[:, -1].values
```

We're using pandas to import our Network Intrusion Dataset into our workspace. Then a set of dependent and independent qualities is defined. The set of independent qualities is labelled X, while the set of dependent attributes is labelled y.

```
In [3]: #Splitting the dataset
        from sklearn.model_selection import train_test_split
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)
```

We're going to divide our dataset into two parts: a training set and a test set. We're going to maintain 25% of the dataset in the test set and 75% in the training set.

```
In [4]: #Fitting our model
        from sklearn.tree import DecisionTreeClassifier
        classifier = DecisionTreeClassifier(criterion = 'entropy' ,random_state = 0)
        classifier.fit (X_train, y_train)

Out[4]: DecisionTreeClassifier(criterion='entropy', random_state=0)
```

We're taking data from the training set to train our model. For our decision tree classifier, we employed "entropy" as the deciding factor.

```
In [5]: #Predicting the Test set Results
        y_pred = classifier.predict(X_test)
```

We're collecting our anticipated test set results from the classifier and saving them in the y pred variable.

```
In [6]: #Printing the confusion matrix
        from sklearn.metrics import confusion_matrix
        cm = confusion_matrix(y_test, y_pred)
        print(cm)

        [[2898   13]
         [  16 3371]]
```

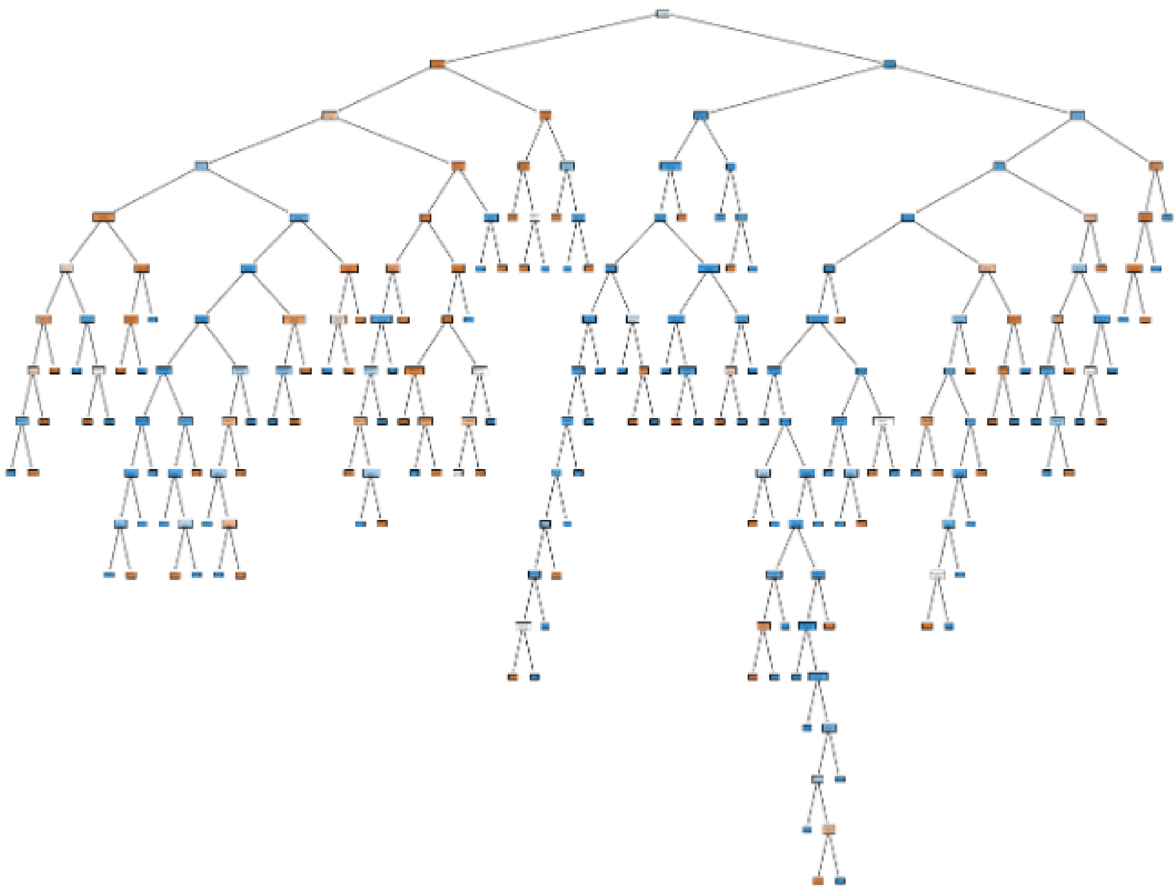
```
In [7]: #Printing the accuracy of our model
        from sklearn.metrics import accuracy_score
        accuracy = accuracy_score(y_test, y_pred)
        print(accuracy)

        0.9953953636074945
```

The confusion matrix and accuracy of our decision tree classifier are printed here. Our model's accuracy with the test dataset is 99.53953636 percent.

```
In [8]: #Defining the labels of our dataset
        classes = ["Anamoly", "Normal"]
```

We're using sklearn's tree library to visualise our decision tree.

In [9]:
```
#Printing the visualized decision tree
fig = plt.figure(figsize=(25,20))
_ = tree.plot_tree(classifier,
                   feature_names=dataset.columns,
                   class_names=classes,
                   filled=True)
```



```
In [10]: #Printing the feature wise break points of our decision tree
test_representation = tree.export_text(classifier)
print(test_representation)
```

```
|--- feature_0 <= 28.50
|   |--- feature_18 <= 8.50
|   |   |--- feature_31 <= 0.50
|   |   |   |--- feature_28 <= 2.50
|   |   |   |   |--- feature_35 <= 0.01
|   |   |   |   |   |--- feature_1 <= 7.50
|   |   |   |   |   |   |--- feature_27 <= 156.00
|   |   |   |   |   |   |   |--- feature_0 <= 0.50
|   |   |   |   |   |   |   |   |--- feature_29 <= 0.51
|   |   |   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |   |   |   |--- feature_29 >  0.51
|   |   |   |   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |   |   |   |--- feature_0 >  0.50
|   |   |   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |   |   |--- feature_27 >  156.00
|   |   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |   |--- feature_1 >  7.50
|   |   |   |   |   |   |--- feature_25 <= 0.75
|   |   |   |   |   |   |   |--- class: normal
```

The categorization criterion of our decision tree is presented here. We can see that feature 0 is our classifier's root node, followed by multiple middle nodes.

Results and Output

Confusion Matrix:

```
[[2898   13]
 [  16 3371]]
```

This is our confusion matrix.

True Negatives: 3488

True Positives: 4032

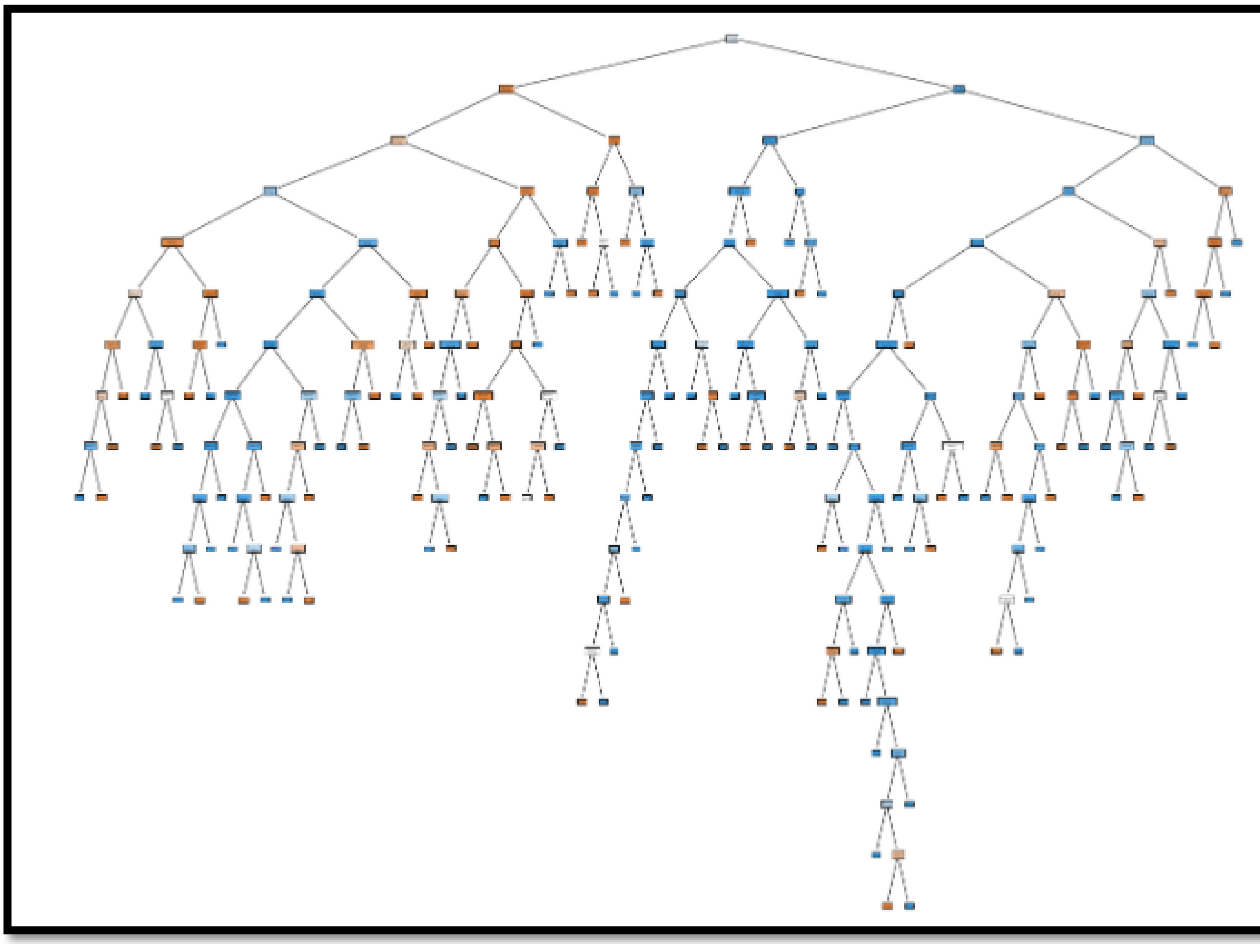
False Positives: 17

False Negatives: 21

Accuracy:

The accuracy of our model stands at 99.49%

Decision Tree Visualization:



Classification Points:

```
|   |   |   |   |   |   |--- feature_27 >  156.00
|   |   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |   |--- feature_1 >  7.50
|   |   |   |   |   |   |--- feature_25 <= 0.75
|   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |   |--- feature_25 >  0.75
|   |   |   |   |   |   |   |--- feature_1 <= 122.50
|   |   |   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |   |   |   |--- feature_1 >  122.50
|   |   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |--- feature_35 >  0.01
|   |   |   |   |   |--- feature_1 <= 201.00
|   |   |   |   |   |   |--- feature_29 <= 0.59
|   |   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |   |   |--- feature_29 >  0.59
|   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |--- feature_1 >  201.00
|   |   |   |   |   |   |--- class: normal
|   |   |   |--- feature_28 >  2.50
|   |   |   |   |--- feature_33 <= 0.87
```

```
|   |   |   |   |--- feature_33 <= 0.87
|   |   |   |   |   |--- feature_30 <= 0.57
|   |   |   |   |   |   |--- feature_18 <= 3.50
|   |   |   |   |   |   |   |--- feature_30 <= 0.13
|   |   |   |   |   |   |   |   |--- feature_29 <= 0.03
|   |   |   |   |   |   |   |   |   |--- feature_1 <= 18.00
|   |   |   |   |   |   |   |   |   |   |--- feature_28 <= 6.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |   |   |   |   |   |--- feature_28 >  6.00
|   |   |   |   |   |   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |   |   |   |   |   |--- feature_1 >  18.00
|   |   |   |   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |   |   |   |--- feature_29 >  0.03
|   |   |   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |   |   |--- feature_30 >  0.13
|   |   |   |   |   |   |   |   |--- feature_28 <= 135.00
|   |   |   |   |   |   |   |   |   |--- feature_27 <= 119.00
|   |   |   |   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |   |   |   |   |--- feature_27 >  119.00
|   |   |   |   |   |   |   |   |   |   |--- feature_30 <= 0.36
```

```
|   |   |   |   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |   |   |   |   |--- feature_27 >  119.00
|   |   |   |   |   |   |   |   |   |   |--- feature_30 <= 0.36
|   |   |   |   |   |   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |   |   |   |   |   |   |--- feature_30 >  0.36
|   |   |   |   |   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |   |   |   |--- feature_28 >  135.00
|   |   |   |   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |   |   |--- feature_18 >  3.50
|   |   |   |   |   |   |   |--- feature_32 <= 0.01
|   |   |   |   |   |   |   |   |--- feature_31 <= 0.01
|   |   |   |   |   |   |   |   |   |--- feature_30 <= 0.05
|   |   |   |   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |   |   |   |   |--- feature_30 >  0.05
|   |   |   |   |   |   |   |   |   |   |--- feature_27 <= 109.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |   |   |   |   |   |--- feature_27 >  109.50
|   |   |   |   |   |   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |   |   |   |   |--- feature_31 >  0.01
|   |   |   |   |   |   |   |   |   |--- class: anomaly
```

```
|   |   |   |   |   |   |   |--- feature_32 >  0.01
|   |   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |--- feature_30 >  0.57
|   |   |   |   |   |   |--- feature_35 <= 0.25
|   |   |   |   |   |   |   |--- feature_32 <= 0.50
|   |   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |   |   |--- feature_32 >  0.50
|   |   |   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |   |   |--- feature_35 >  0.25
|   |   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |--- feature_33 >  0.87
|   |   |   |   |   |--- feature_34 <= 0.11
|   |   |   |   |   |   |--- feature_30 <= 0.06
|   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |   |--- feature_30 >  0.06
|   |   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |   |--- feature_34 >  0.11
|   |   |   |   |   |   |--- class: anomaly
|   |   |--- feature_31 >  0.50
|   |   |   |--- feature_7 <= 0.50
```

```
|   |   |--- feature_31 >  0.50
|   |   |   |--- feature_7 <= 0.50
|   |   |   |   |--- feature_0 <= 0.50
|   |   |   |   |   |--- feature_27 <= 4.00
|   |   |   |   |   |   |--- feature_36 <= 0.10
|   |   |   |   |   |   |   |--- feature_28 <= 179.50
|   |   |   |   |   |   |   |   |--- feature_20 <= 0.25
|   |   |   |   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |   |   |   |   |--- feature_20 >  0.25
|   |   |   |   |   |   |   |   |   |--- feature_32 <= 0.36
|   |   |   |   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |   |   |   |   |--- feature_32 >  0.36
|   |   |   |   |   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |   |   |   |--- feature_28 >  179.50
|   |   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |   |--- feature_36 >  0.10
|   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |--- feature_27 >  4.00
|   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |--- feature_0 >  0.50
```

```
|   |   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |   |   |--- feature_0 >  1133.50
|   |   |   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |   |   |--- feature_30 >  0.00
|   |   |   |   |   |   |   |--- class: normal
|   |   |   |   |--- feature_0 >  6052.00
|   |   |   |   |   |--- class: anomaly
|   |   |--- feature_18 >  42.50
|   |   |   |--- feature_1 <= 2.00
|   |   |   |   |--- feature_7 <= 0.50
|   |   |   |   |   |--- feature_31 <= 0.00
|   |   |   |   |   |   |--- class: normal
|   |   |   |   |   |--- feature_31 >  0.00
|   |   |   |   |   |   |--- class: anomaly
|   |   |   |   |--- feature_7 >  0.50
|   |   |   |   |   |--- class: normal
|   |   |   |--- feature_1 >  2.00
|   |   |   |   |--- class: normal
```