**109550198** 卜銳凱

**Introduction to Data Science**

HW5

# Cyber Security Attacks:

Consists of 25 varied metrics and 40,000 records

About the dataset:

Welcome to Incribo's synthetic tourism dataset! Crafted with precision, this dataset offers a realistic representation of travel history, making it an ideal playground for various analytical tasks.
Use the cybersecurity attacks dataset to help you assess the heatmaps, attack signatures, types, and more!

This code is a Python script that appears to be analyzing a dataset related to cybersecurity attacks. Here is a summary of what the code does and a brief report:

**Importing Libraries**: The script begins by importing Python libraries such as NumPy, matplotlib, pandas, and seaborn. These libraries are often used to manipulate and visualize data.

**Data Loading**: Using the Pandas library, the script reads a CSV file entitled 'cybersecurity_attacks.csv' and stores it in a DataFrame named dataset. The information is divided into characteristics (x) and the target variable (y).

**Data Exploration**: The code performs a variety of data exploration and analysis activities, including the following: It uses dataset.head(10) to print the first ten rows of the dataset and type(dataset) to display the data types of the objects.

- It looks for duplicate rows in the dataset and outputs the number of duplicate rows.
- It removes duplicate rows from the dataset to ensure data integrity.
- It displays the number of rows and columns in the dataset, as well as the column names. Using dataset.describe(), it returns summary statistics for the dataset.
- It uses dataset.info () to inspect the dataset's structure and prints unique values for each column.

**Handling Missing Data**:

The script checks the dataset for missing values and prints the number of missing values for each column.
It also generates a heatmap to display missing data in the dataset.
It computes the proportion of missing data for each column and prints the results.

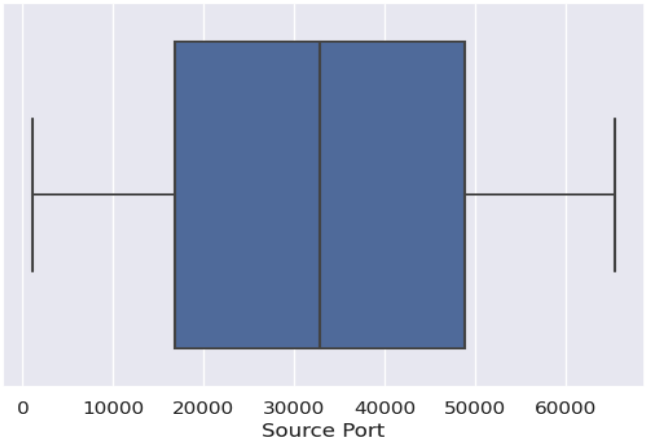
**Data Visualization**:

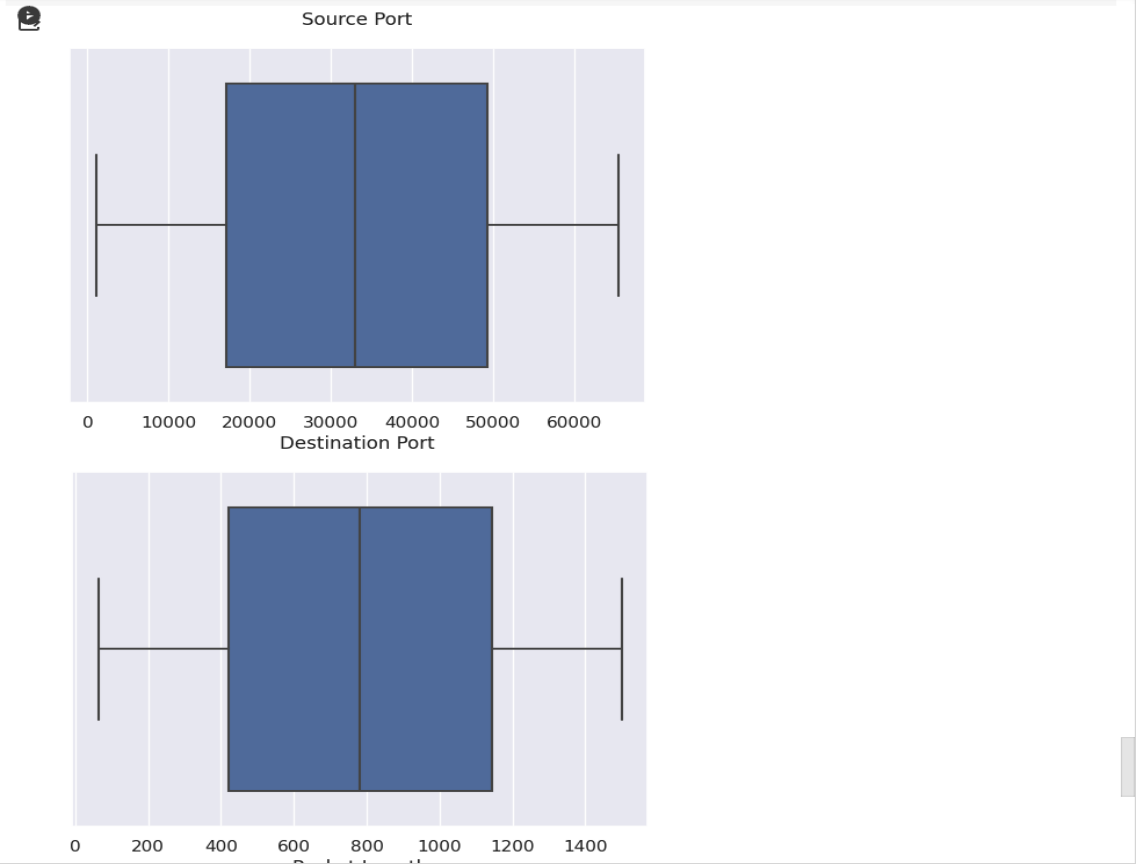
The code uses a for loop to create box plots for columns with numeric data types (int64 and float64). It identifies whether there are outliers or not. After observation, there are no outliers in the dataset.

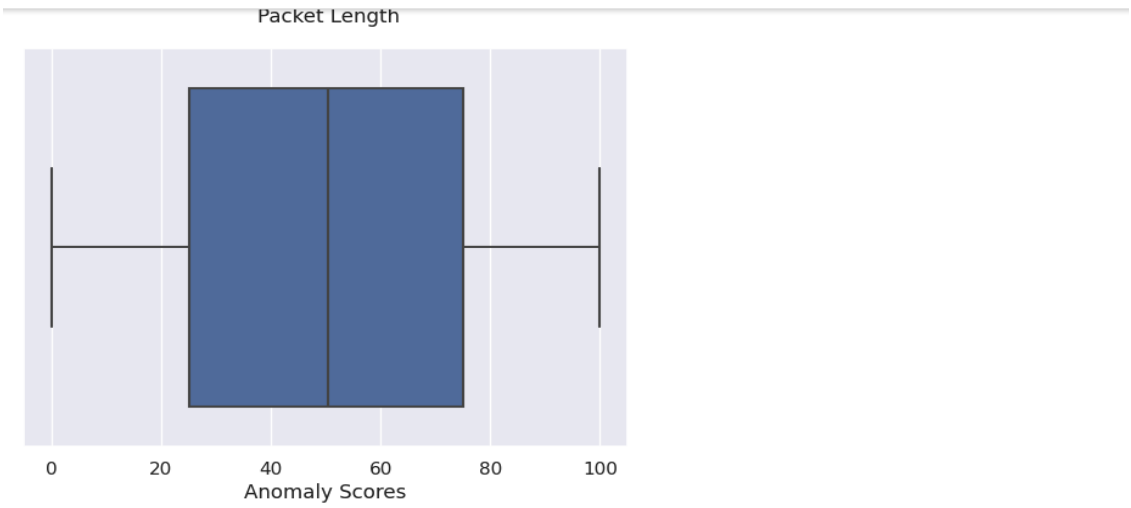
## Detecting outliers

```
# for col_name in dataset.columns:
data = dataset.dtypes
for i, (key , value) in enumerate (data.items()):
  if value.name == 'int64' or value.name== 'float64':
    plt.figure(i)
    sns.boxplot(x=dataset[key])
    print(key)
    print(value)
```

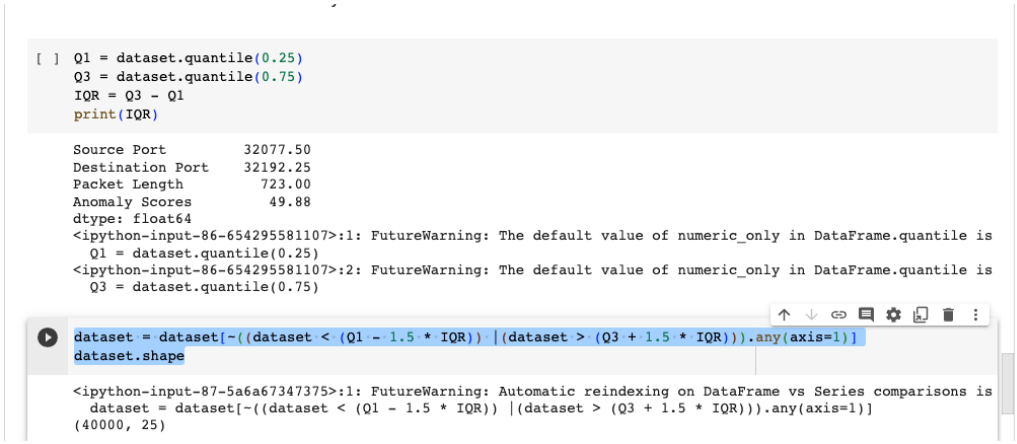
```
Source Port
int64
Destination Port
int64
Packet Length
int64
Anomaly Scores
float64
```






To make sure there are no outliers in the dataset, we use the Interquartile Range (IQR) method and removes rows with outliers. After observation, there are no outliers in the dataset.



```
[ ] Q1 = dataset.quantile(0.25)
    Q3 = dataset.quantile(0.75)
    IQR = Q3 - Q1
    print(IQR)

    Source Port         32077.50
    Destination Port    32192.25
    Packet Length         723.00
    Anomaly Scores         49.88
    dtype: float64
    <ipython-input-86-654295581107>:1: FutureWarning: The default value of numeric_only in DataFrame.quantile is
      Q1 = dataset.quantile(0.25)
    <ipython-input-86-654295581107>:2: FutureWarning: The default value of numeric_only in DataFrame.quantile is
      Q3 = dataset.quantile(0.75)

    dataset = dataset[~((dataset < (Q1 - 1.5 * IQR)) |(dataset > (Q3 + 1.5 * IQR))).any(axis=1)]
    dataset.shape

    <ipython-input-87-5a6a67347375>:1: FutureWarning: Automatic reindexing on DataFrame vs Series comparisons is
      dataset = dataset[~((dataset < (Q1 - 1.5 * IQR)) |(dataset > (Q3 + 1.5 * IQR))).any(axis=1)]
    (40000, 25)
```

Overall, this code is a data exploration and preprocessing script for a cybersecurity attacks dataset. It aims to clean the data by handling missing values and outliers. It also provides some initial insights into the dataset using summary statistics and data visualization.