Teachnook Final Project Feb-Batch

import pandas as pd

import numpy as np

import random

from faker import Faker

from datetime import datetime, timedelta

import matplotlib.pyplot as plt

import seaborn as sns

fake = Faker()

def generate\_network\_traffic\_data(num\_samples):

    data = []

    anomaly\_types = ['Normal', 'Intrusion Attempt', 'Port Scanning', 'DDoS Attack']

    for \_ in range(num\_samples):

        timestamp = fake.date\_time\_between(start\_date='-1d', end\_date='now')

        source\_ip = fake.ipv4()

        destination\_ip = fake.ipv4()

        source\_port = random.randint(1024, 65535)

        destination\_port = random.randint(1, 1023)

        protocol = random.choice(['TCP', 'UDP'])

        packet\_length = random.randint(64, 1500)

        payload = fake.text(max\_nb\_chars=random.randint(10, 100))

        anomaly\_type = random.choice(anomaly\_types)

        label = 1 if anomaly\_type != 'Normal' else 0

        data.append([timestamp, source\_ip, destination\_ip, source\_port, destination\_port, protocol,

                     packet\_length, payload, anomaly\_type, label])

    return data

num\_samples = 10

# Generate synthetic network traffic data

network\_traffic\_data = generate\_network\_traffic\_data(num\_samples)

columns = ['Timestamp', 'Source IP', 'Destination IP', 'Source Port', 'Destination Port',

           'Protocol', 'Packet Length', 'Payload', 'Anomaly Type', 'Label']

df = pd.DataFrame(network\_traffic\_data, columns=columns)

print("Missing values:\n", df.isnull().sum())

print("\nSummary statistics:")

print(df.describe())

num\_anomalies = df[df['Anomaly Type'] != 'Normal'].shape[0]

print("\nNumber of anomalies:", num\_anomalies)

print("\nDistribution of labels:")

print(df['Label'].value\_counts())

print("\nDistribution of anomaly types:")

print(df['Anomaly Type'].value\_counts())

print("\nDetails of anomalies:")

anomalies = df[df['Anomaly Type'] != 'Normal']

for idx, row in anomalies.iterrows():

    print(f"Alert: Suspicious activity detected - {row['Anomaly Type']} at {row['Timestamp']}. "

          f"Source IP: {row['Source IP']}, Destination IP: {row['Destination IP']}, "

          f"Source Port: {row['Source Port']}, Destination Port: {row['Destination Port']}, "

          f"Protocol: {row['Protocol']}, Packet Length: {row['Packet Length']}, Payload: {row['Payload']}")

plt.figure(figsize=(16, 10))

plt.subplot(2, 2, 1)

sns.countplot(x='Protocol', data=df)

plt.xlabel('Protocol')

plt.ylabel('Count')

plt.title('Network Traffic by Protocol')

plt.subplot(2, 2, 2)

sns.histplot(df['Packet Length'], bins=30, kde=True)

plt.xlabel('Packet Length')

plt.ylabel('Frequency')

plt.title('Distribution of Packet Length')

plt.subplot(2, 2, 3)

plt.scatter(df['Packet Length'], df['Source Port'], c=df['Label'], cmap='viridis', alpha=0.5)

plt.xlabel('Packet Length')

plt.ylabel('Source Port')

plt.title('Packet Length vs Source Port')

plt.subplot(2, 2, 4)

plt.scatter(df['Packet Length'], df['Destination Port'], c=df['Label'], cmap='viridis', alpha=0.5)

plt.xlabel('Packet Length')

plt.ylabel('Destination Port')

plt.title('Packet Length vs Destination Port')

plt.tight\_layout()

plt.show()