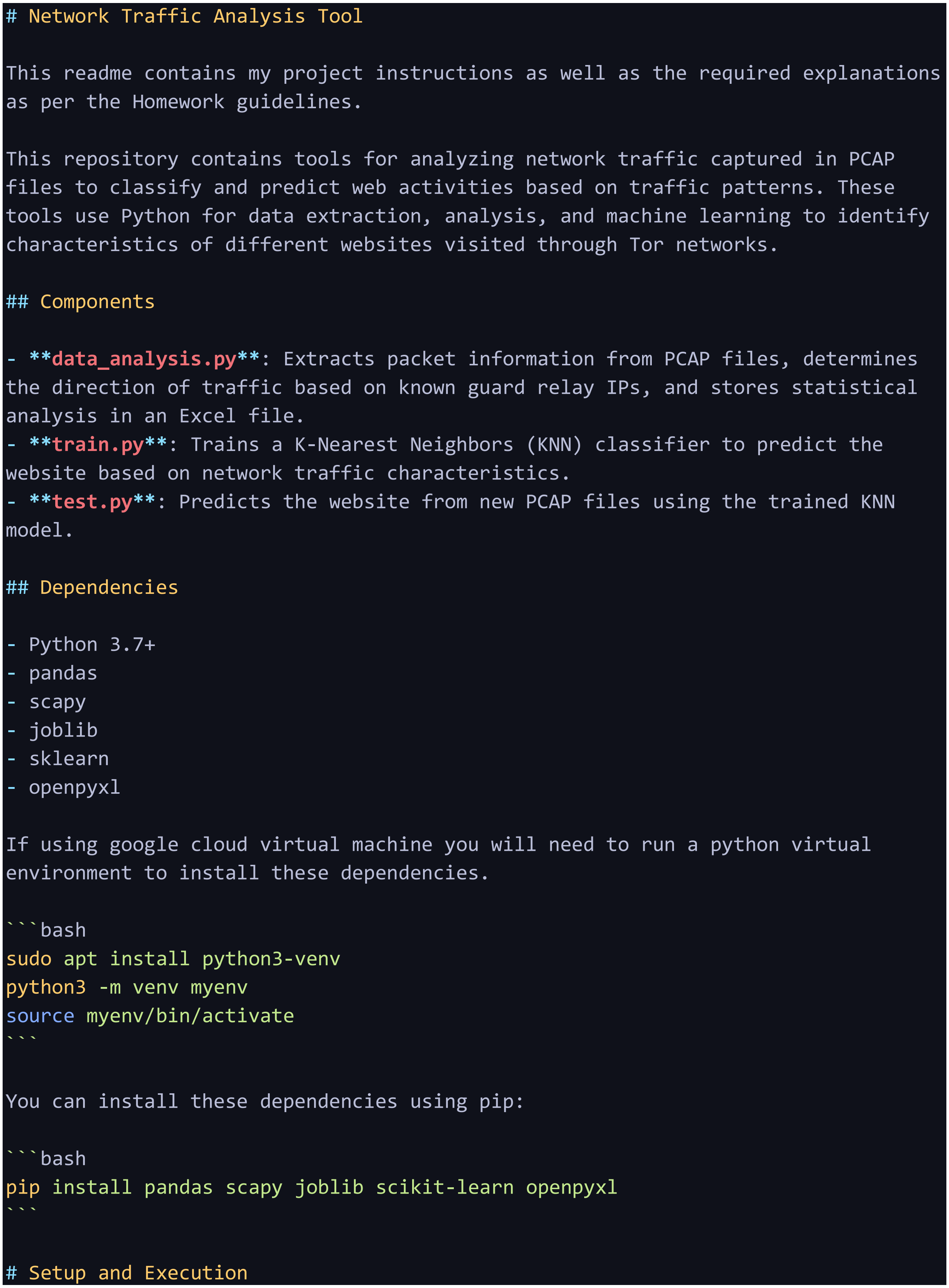


# Network Traffic Analysis Tool

This readme contains my project instructions as well as the required explanations as per the Homework guidelines.

This repository contains tools for analyzing network traffic captured in PCAP files to classify and predict web activities based on traffic patterns. These tools use Python for data extraction, analysis, and machine learning to identify characteristics of different websites visited through Tor networks.

## Components

- **data_analysis.py**: Extracts packet information from PCAP files, determines the direction of traffic based on known guard relay IPs, and stores statistical analysis in an Excel file.
- **train.py**: Trains a K-Nearest Neighbors (KNN) classifier to predict the website based on network traffic characteristics.
- **test.py**: Predicts the website from new PCAP files using the trained KNN model.

## Dependencies

- Python 3.7+
- pandas
- scapy
- joblib
- sklearn
- openpyxl

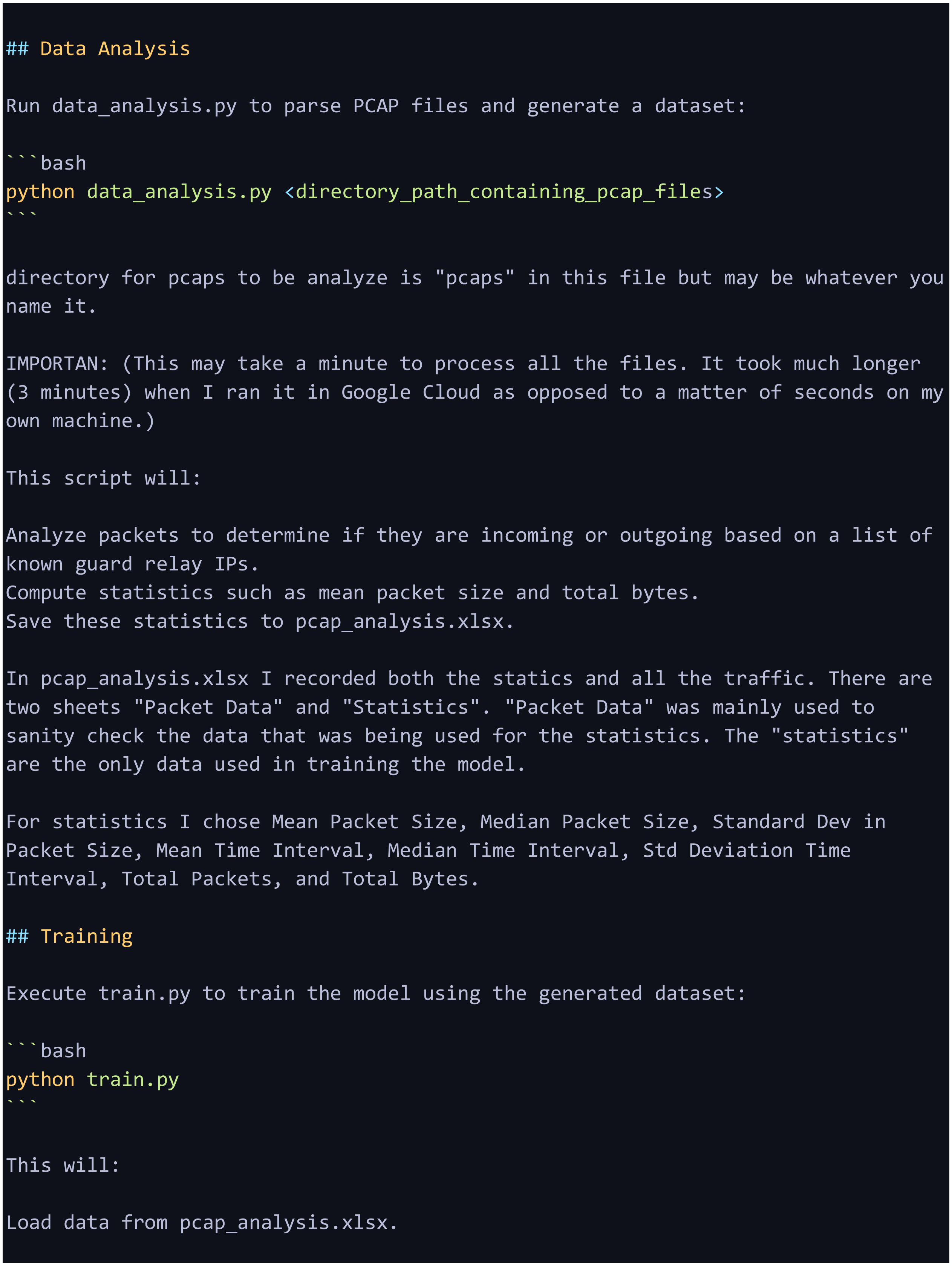
If using google cloud virtual machine you will need to run a python virtual environment to install these dependencies.

```bash
sudo apt install python3-venv
python3 -m venv myenv
source myenv/bin/activate
```

You can install these dependencies using pip:

```bash
pip install pandas scapy joblib scikit-learn openpyxl
```

# Setup and Execution



## Data Analysis

Run data_analysis.py to parse PCAP files and generate a dataset:

```bash
python data_analysis.py <directory_path_containing_pcap_files>
```

directory for pcaps to be analyze is "pcaps" in this file but may be whatever you name it.

IMPORTAN: (This may take a minute to process all the files. It took much longer (3 minutes) when I ran it in Google Cloud as opposed to a matter of seconds on my own machine.)

This script will:

Analyze packets to determine if they are incoming or outgoing based on a list of known guard relay IPs.
Compute statistics such as mean packet size and total bytes.
Save these statistics to pcap_analysis.xlsx.

In pcap_analysis.xlsx I recorded both the statics and all the traffic. There are two sheets "Packet Data" and "Statistics". "Packet Data" was mainly used to sanity check the data that was being used for the statistics. The "statistics" are the only data used in training the model.

For statistics I chose Mean Packet Size, Median Packet Size, Standard Dev in Packet Size, Mean Time Interval, Median Time Interval, Std Deviation Time Interval, Total Packets, and Total Bytes.

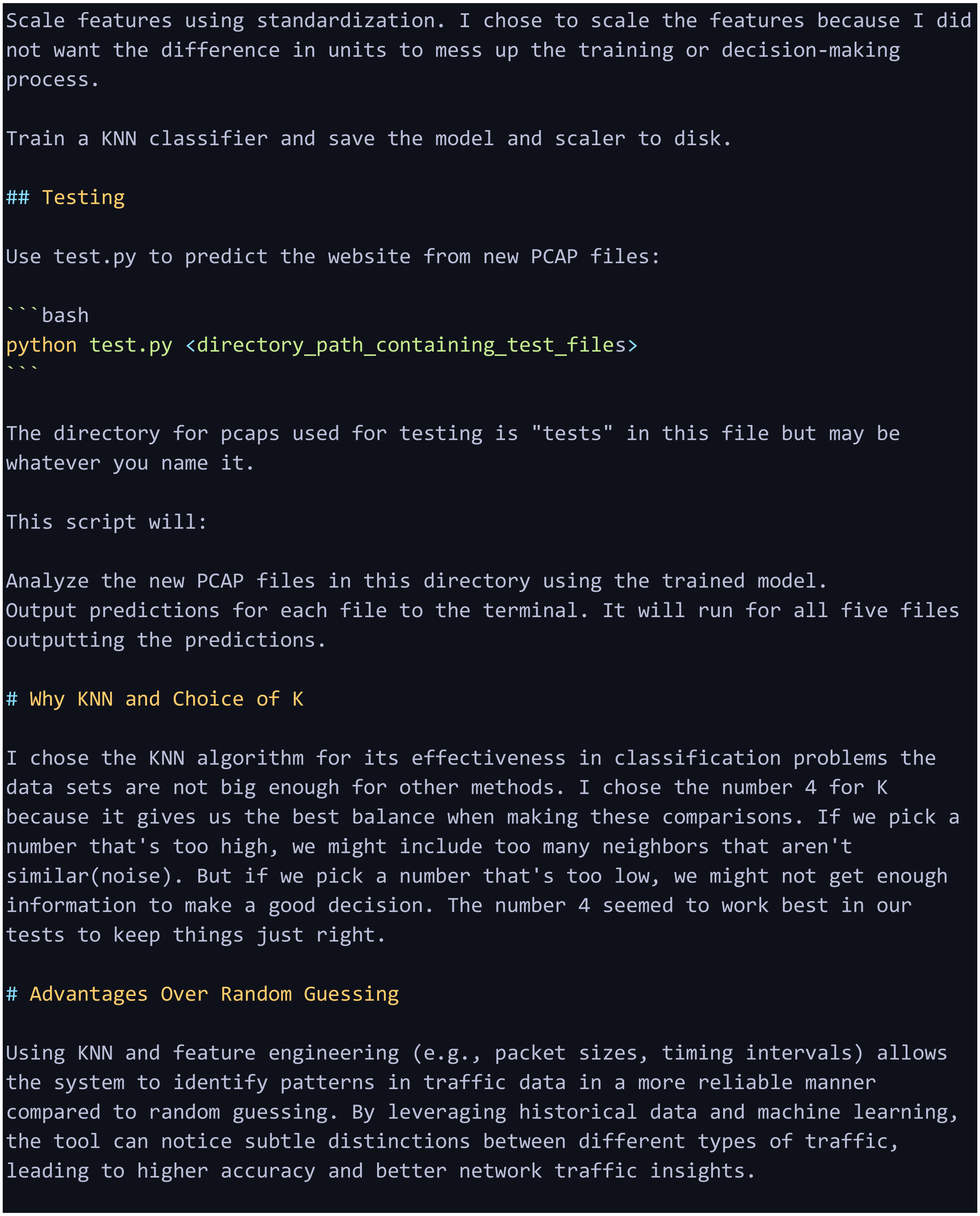
## Training

Execute train.py to train the model using the generated dataset:

```bash
python train.py
```

This will:

Load data from pcap_analysis.xlsx.



~~~markdown
Scale features using standardization. I chose to scale the features because I did
not want the difference in units to mess up the training or decision-making
process.

Train a KNN classifier and save the model and scaler to disk.

## Testing

Use test.py to predict the website from new PCAP files:

```bash
python test.py <directory_path_containing_test_files>
```

The directory for pcaps used for testing is "tests" in this file but may be
whatever you name it.

This script will:

Analyze the new PCAP files in this directory using the trained model.
Output predictions for each file to the terminal. It will run for all five files
outputting the predictions.

# Why KNN and Choice of K

I chose the KNN algorithm for its effectiveness in classification problems the
data sets are not big enough for other methods. I chose the number 4 for K
because it gives us the best balance when making these comparisons. If we pick a
number that's too high, we might include too many neighbors that aren't
similar(noise). But if we pick a number that's too low, we might not get enough
information to make a good decision. The number 4 seemed to work best in our
tests to keep things just right.

# Advantages Over Random Guessing

Using KNN and feature engineering (e.g., packet sizes, timing intervals) allows
the system to identify patterns in traffic data in a more reliable manner
compared to random guessing. By leveraging historical data and machine learning,
the tool can notice subtle distinctions between different types of traffic,
leading to higher accuracy and better network traffic insights.
~~~