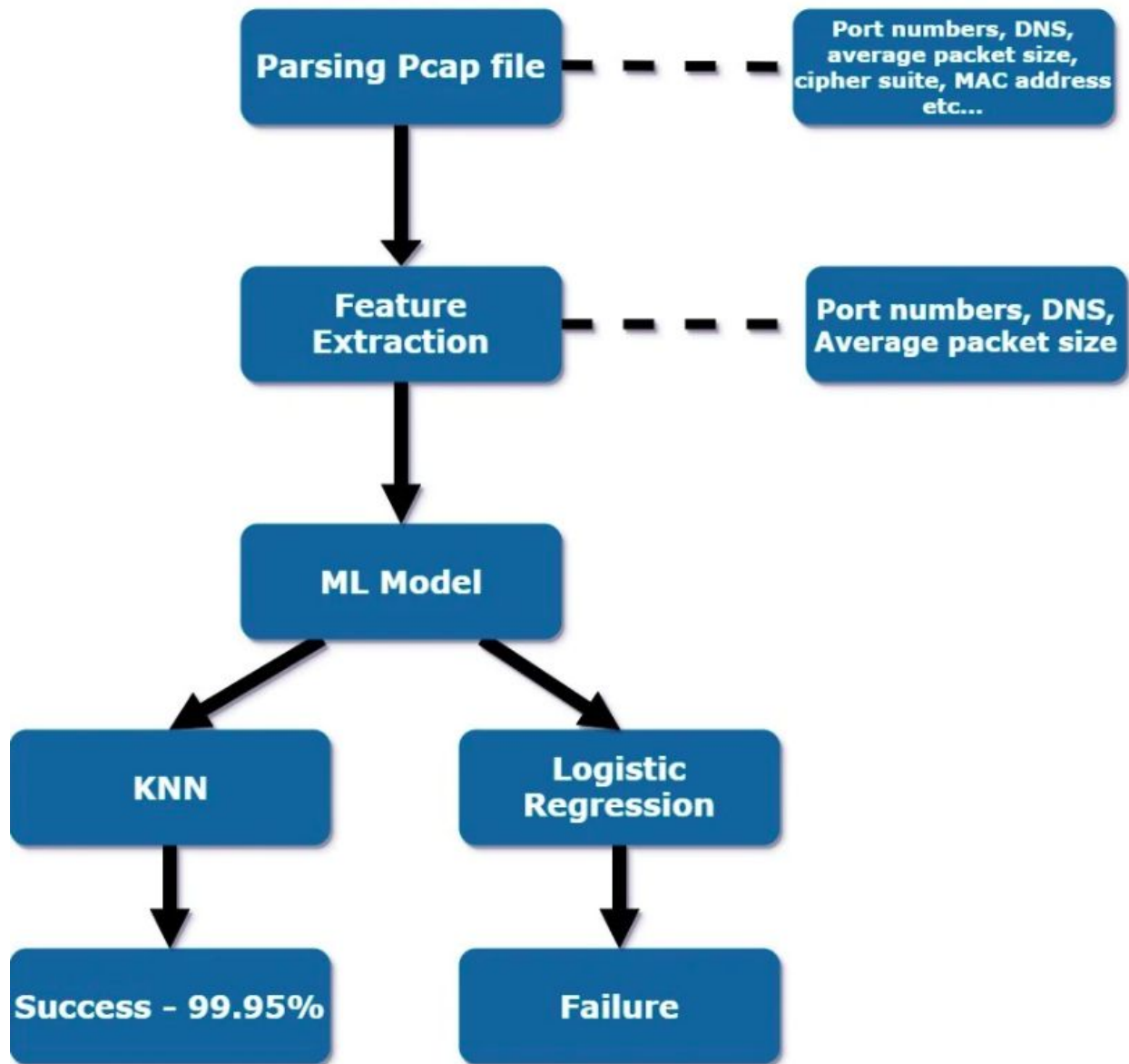


IoT Devices Classification by Analyzing Network Traffic Characteristics →

Author → Lovepreet Singh, Prabhakar Kumar, Vignesh.

Steps to follow for IoT Devices Classification →



- 1.) Parsing → Use Wireshark Tool to convert the Pcap file to CSV because we will use CSV file as a dataset for ML classification. Add Features by using column addition option from the Wireshark tool.

For adding a column in Wireshark so that this column can be the feature in our CSV file, Follow these steps →

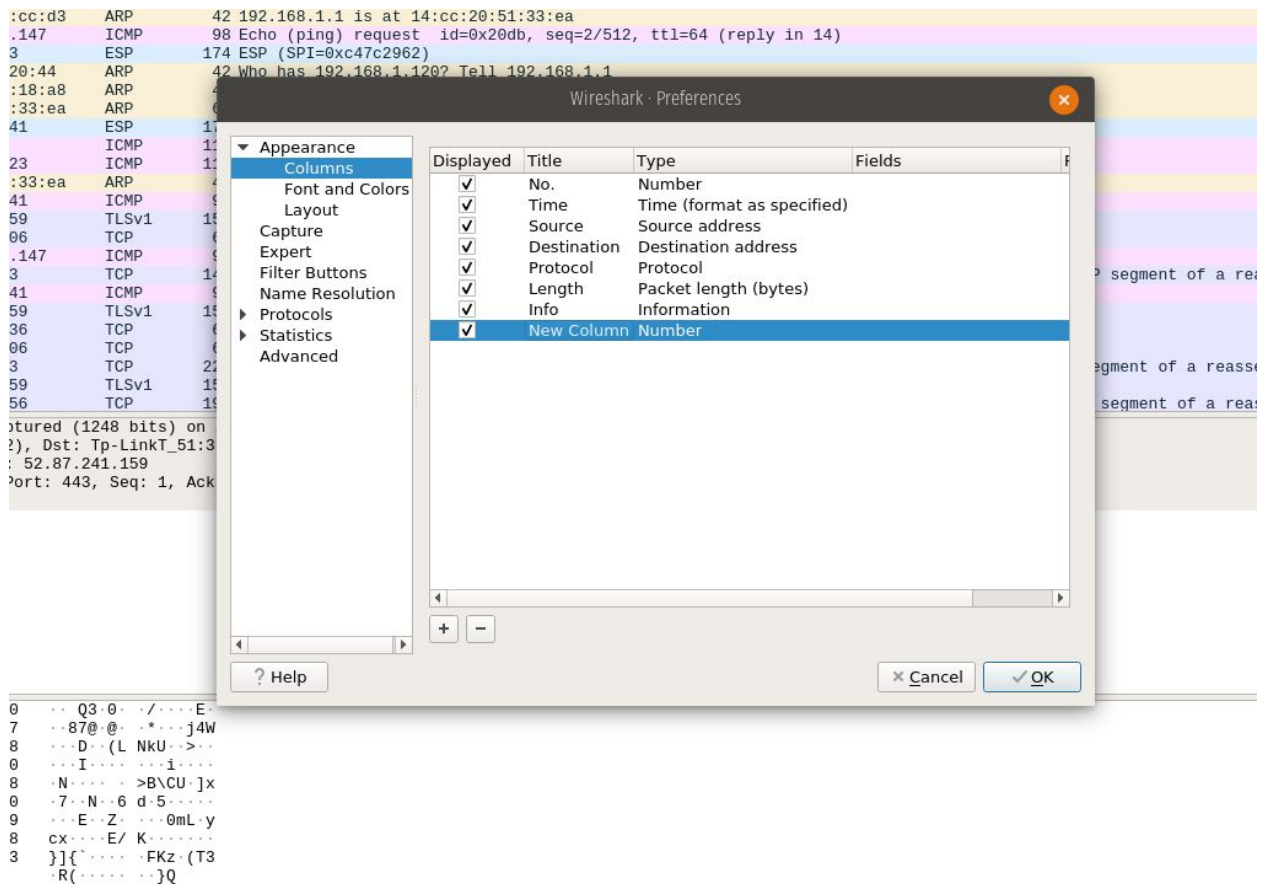
Go to Edit→ Preferences→ Columns and click on add button.

The screenshot displays the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The main window is divided into three panes: Packet List, Packet Details, and Packet Bytes.

Packet List Pane: Shows a list of captured packets. The columns are No., Time, Source, Destination, Protocol, and Length. The first packet (No. 1) is a TCP Reset (RST) from 192.168.1.106 to 52.87.241.159. Subsequent packets show a series of Echo (ping) requests and replies between the same IP addresses, as well as some application data.

Packet Details Pane: Provides a hierarchical view of the selected packet's structure. For the first packet, it shows Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Secure Sockets Layer. For the second packet, it shows the Application Data layer.

Packet Bytes Pane: Displays the raw hexadecimal and ASCII data of the selected packet. The first packet's data is shown as a series of hexadecimal values and their corresponding ASCII characters.



We have added new columns of Ports, Mac Address (For separating IoTs).

- 2.) Use Mac address CSV for separating the features of individual IoT devices and label them.

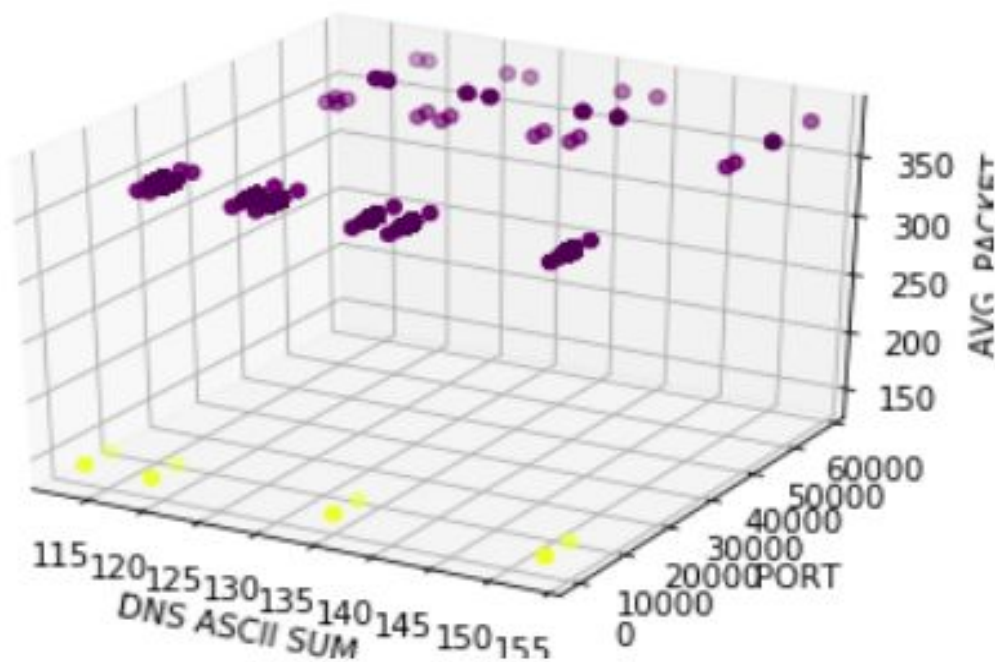
Note: → You do not need to perform preprocessing because This code has all CSV already (Extracted from Wireshark Tool).

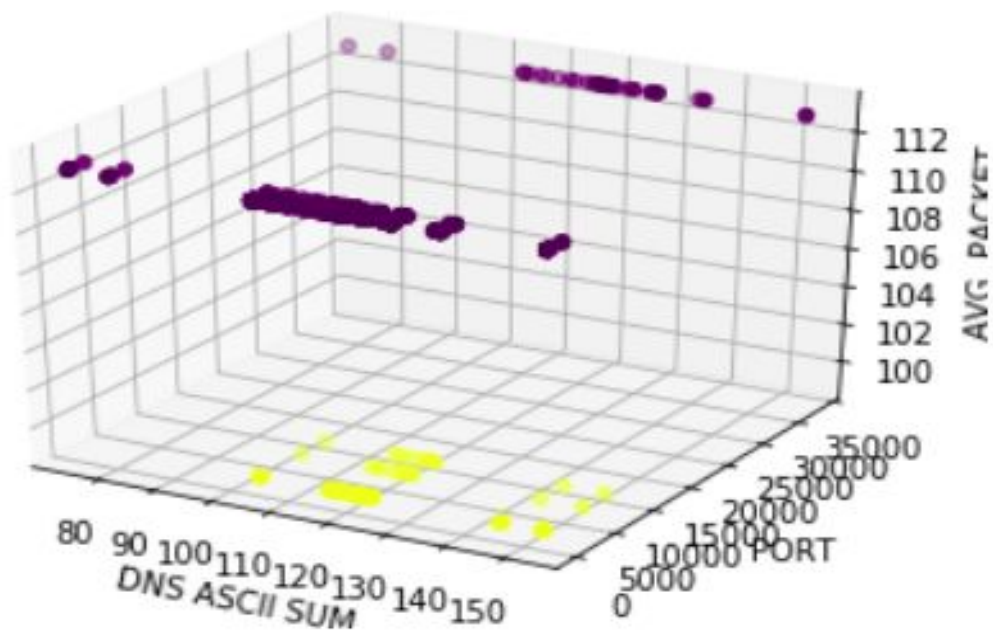
- 3.) After Labelling the dataset, Remove Protocols like ARP and ICMPv6.
- 4.) Now, We have extracted the ports, Dns and Average packet size which is total data sent and received divided by number of packets.
- 5.) And After labeling the dataset, we have analyzed feature importance using data polation and we found that DNS, Average Packet Size and Ports, these three features are enough for classification.
- 6.) We found that these three features are forming dense clusters and thus KNN worked well because it works on the principle of Euclidean Distance.

7.) To cross-check that KNN is the only good algorithm to apply we applied the Logistic Regression which assumes that Dataset is linearly separable. But our dataset is forming dense clusters that are not linearly separable. So we got almost 70% accuracy while applying Logistic Regression and 99.95% with KNN.

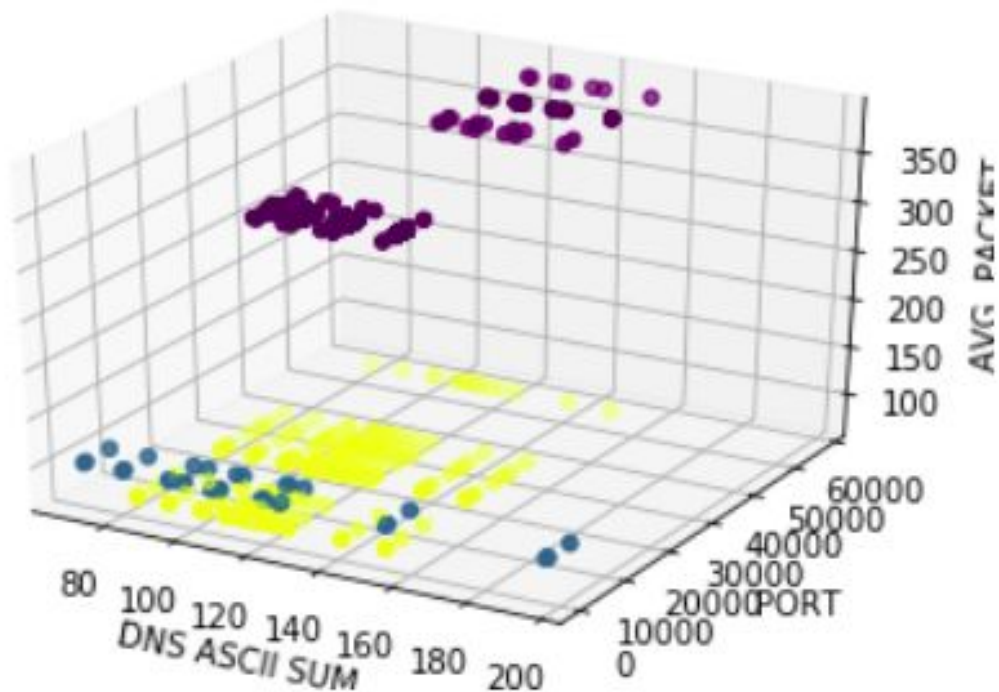
Few Important Points and Reasons →

KNN algorithm performed well and we got about 99.95% accuracy. It is because IoT Devices are forming their dense clusters, And when new datapoint comes for classification it falls in one of the 28-clusters(because we have 28 IoT Devices). KNN works on the principle of Euclidean distance and therefore distance measurement performed well in case of dense clusters of different classes. In fig 2, we can see from the graphs of IoT Devices (2 IoT taken here) that all instances of the dataset are forming dense clusters.

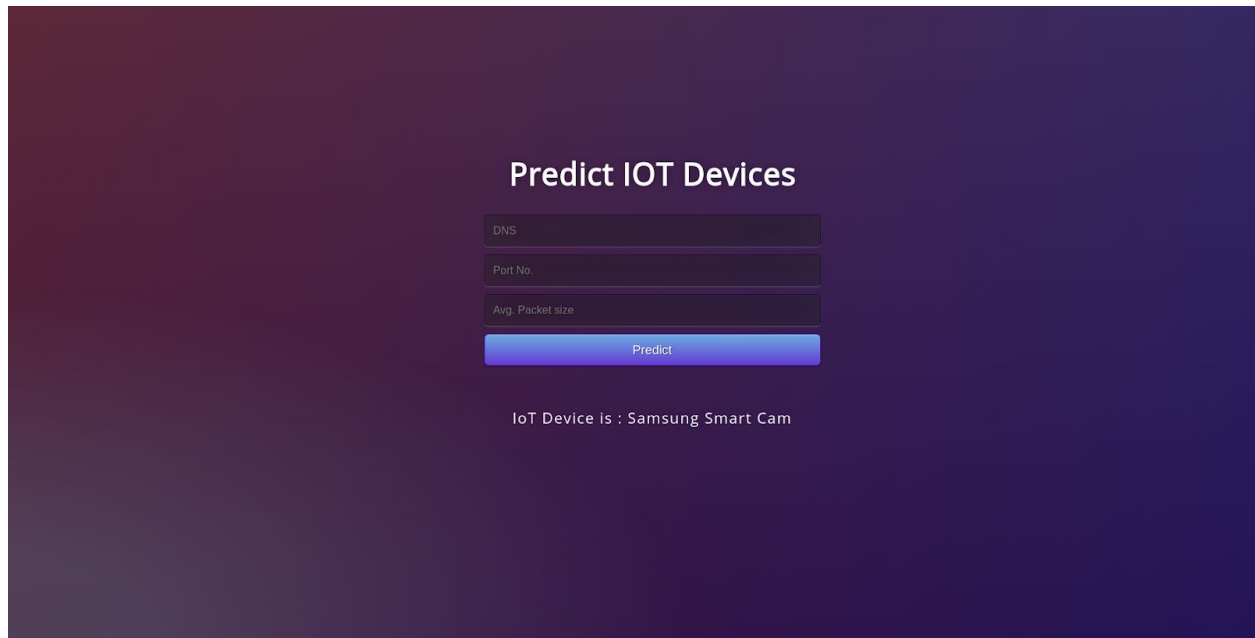




Logistic Regression is not like KNN, it assumes that data is almost linearly separable. This is because it draws linear hyper-plane or decision boundaries for classification. But this dataset is not linearly separable as shown in fig 3. Therefore, Logistic regression is unable to classify this non-linearly separable dataset.



We finally used the FLASK Python Package for User interaction with our ML model Classification.



The screenshot shows a web application interface with a dark purple gradient background. At the top center, the title "Predict IOT Devices" is displayed in white. Below the title are three input fields: "DNS", "Port No.", and "Avg. Packet size", each with a light gray border. A blue "Predict" button is positioned below these fields. At the bottom center, the text "IoT Device is : Samsung Smart Cam" is shown in white.

ENJOY WITH Code</> :)