DATA TO EXTRACT FROM PCAP FILES:

CATEGORY	WHAT TO EXTRACT	WHAT IT REPRESENTS	WHAT IT IS FOR (Relating to Project)
Basic Packet Info	**Timestamp**	Time packet was captured (seconds since epoch).	Used to calculate **flow duration**, **plot traffic over time**, and detect bursts.
	Source IP Address	IP address of sender.	Identifies **user device** or local source; helps separate **upload/download** traffic.
	Destination IP Address	IP address of receiver.	Identifies **remote server** (e.g., YouTube, Google); helps find **which services were accessed**.
	Source Port	Sending application's port number.	Distinguishes **application type** (e.g., 443 = HTTPS).
	Destination Port	Receiving application's port number.	Identifies **service type** (e.g., 53 = DNS, 443 = HTTPS, QUIC = YouTube).
	Protocol	Transport or application-level protocol.	Used to **categorize traffic type** (streaming, browsing, DNS, etc.).
	Packet Length	Total size of packet (in bytes).	Used for **average/variance of packet size**; helps detect **streaming vs browsing**.
	Packet Direction	Whether the packet is outgoing or incoming.	Helps calculate **upload/download ratios** for traffic classification.
	Info Field	Packet content summary (e.g., ACK, Application Data).	Useful for **identifying TCP handshakes** and **flow stages**.
Flow-Level Info (Aggregated per Connection)	**Flow ID (5-tuple)**	Unique identifier for a single session or communication.	Used to **group packets** into logical connections ("flows") for feature computation.
	Flow Start Time	Time of first packet in flow.	Used for **timing** and **chronological visualization**.
	Flow End Time	Time of last packet in flow.	Used to compute **flow duration**.
	Flow Duration	How long a session lasted.	Distinguishes **short browsing bursts** from **long streaming sessions**.
	Number of Packets (per flow)	Total packets exchanged in flow.	Indicates **session intensity**; higher counts = heavier usage (streaming/download).
	Total Bytes (per flow)	Total amount of data transferred.	Used for **throughput measurement** and **traffic volume visualization**.
	Average Packet Size	Mean packet size per flow.	Helps identify **streaming (large, consistent packets)** vs **browsing (variable packets)**.
	Packets per Second (Rate)	Packet transmission rate.	Used to measure **flow intensity**; streaming shows steady rate.
	Direction Ratio (Upload/Download)	Ratio of outgoing to incoming traffic.	Distinguishes **upload-heavy (video call)** vs **download-heavy (streaming)**.
	Inter-Arrival Time	Time gap between packets.	Reveals **bursty vs continuous** activity (browsing vs streaming).
Transport Layer Info (TCP/UDP)	**TCP Flags (SYN/ACK/FIN)**	Control bits marking connection start/end.	Used to **identify connection boundaries** for flows.
	TCP Retransmissions	Indicates packet loss or retries.	Helps measure **network quality**.
	TCP Window Size	Receiver buffer capacity.	Indicates **congestion or flow control**.
	UDP Length	Length of UDP payload.	Used to analyze **UDP traffic** (e.g., QUIC-based streaming).
Application Layer Info	**TLS Version**	Encryption protocol version (1.2 / 1.3).	Identifies **secure HTTPS sessions**.
	TLS Server Name (SNI)	Domain name being accessed.	Used to **map traffic to specific websites** (e.g., youtube.com).
	QUIC Connection ID	Connection ID for QUIC sessions.	Identifies **YouTube/Google streaming traffic**.
	HTTP Method	Type of HTTP request (GET, POST).	Distinguishes **browsing (GET)** vs **upload (POST)**.
	HTTP Host	Target web host in HTTP header.	Used to **detect websites** and **group traffic by domain**.
	DNS Query Name	Domain requested by user.	Maps **IP addresses to website names** for readability.
Statistical / Behavioral Features (Derived)	**Unique Protocol Count**	Variety of protocols used in session.	Indicates **multi-tasking or background activity**.
	Entropy of Packet Sizes	Variability in packet size.	**Low variance = streaming**, **high = browsing**.
	Bytes per Second	Average throughput.	Shows **bandwidth consumption per flow**.
	Packets per Second	Transmission rate.	Measures **activity intensity**.
	Ratio of Upload to Download Bytes	Upload/download asymmetry.	Distinguishes **upload-heavy vs download-heavy** sessions.
	Number of Active Flows	Total simultaneous connections.	Shows **network concurrency** or multitasking.
Advanced ML Features (Optional)	**Client-to-Server Ratio**	Upload/download relationship.	Used by ML model to separate **chatting vs streaming**.
	Burstiness Index	Packet spacing irregularity.	Detects **real-time vs buffered** traffic.
	Num of Unique Domains	How many different sites accessed.	**Browsing = many sites**, **streaming = few**.
Dashboard & Analysis Focus	**Protocol Distribution**	Overall traffic composition.	Used for **pie chart** showing proportion of protocols.
	Top Talkers (IPs with most bytes)	Devices sending/receiving most data.	Highlights **heavy users or active apps** on dashboard.
	Traffic Volume Over Time	Aggregated time-series data.	Used for **line chart** showing bursts or spikes in activity.
	Flow Table View	Tabular display of flows.	Allows users to **inspect details** of connections.
	Classified Activity Label	From ML output	Translates technical metrics into **user-understandable categories**.

WHAT TO PAY ATTENTION TO (when extracting):

Data-Level

- **Time synchronization:** Ensure PCAP timestamps are correct, consistent, and sorted. Needed for flow segmentation and time-based visualizations.
- **Encryption:** TLS/QUIC hides payloads; analyze metadata (packet sizes, timing) instead of content.
- **Noise / Background traffic:** Filter unrelated traffic by destination IPs (exclude OS updates, background apps).
- **Bot vs human patterns:** Bots often have fixed interval requests; human traffic is irregular.

Feature-Level

- **Flows vs packets:** Aggregate packets per flow; avoid analyzing individual packets.
- **Sampling:** Trim long captures to speed up testing. Decide segmentation windows (per flow, per minute) for graphing and analysis.
- **Missing protocols:** Wireshark may fail to decode new protocols; fallback to port numbers (e.g., 443 → HTTPS).

PCAP-Specific Focus Areas

- **Protocol column:** Shows traffic layer (TCP, UDP, QUIC, TLS). Different activities use different protocols.
- **Port numbers:** Hint at application type (443 → HTTPS, 80 → HTTP, 53 → DNS).
- **Flow duration & packet size:** Help classify activity type (streaming = long/large; browsing = short/bursty).
- **Packet timing (Time column):** Used to calculate inter-arrival times and flow duration.
- **Server names (TLS/HTTP host):** Map traffic to real-world services.
- **QUIC/TLS handshake packets:** Contain metadata even for encrypted sessions (e.g., SNI).
- **Directionality:** Compare `ip.src` to your machine's IP to separate upload vs download.

Scikit-learn

- python library for machine learning built on numpy pandas and matplotlib

machine learning in sklearn: pipeline		
	Step	**Example**
1	Import libraries	from sklearn.linear_model import LinearRegression
2	Load data	X, y = ...
3	Split data	train_test_split()
4	Train model	model.fit(X_train, y_train)
5	Evaluate	model.score(X_test, y_test)

1. **Data Preprocessing:**
   Before using data in ML model:
- Handle missing values
- Convert text to numbers
- Scale features

2. **Splitting Data**

You usually split data into:

- Training set (to teach the model)
- Test set (to check performance on unseen data)

What: You split your table into:

- Features (X): the inputs (age, height, pixel values, etc.)
- Target (y): what you want to predict (price, class label, etc.)

Tip : Always inspect your data (print(head) in pandas). Ask: *Does this column help predict the target?* If not, maybe drop it.*

Example: (https://scikit-learn.org/stable/api/sklearn.model_selection.html)

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

3. **Decide on Model Category**

Example:
Regression: from sklearn.linear_model import LinearRegression
Classification: from sklearn.ensemble import RandomForestClassifier
Clustering: from sklearn.cluster **import** KMeans
Dimensionality Reduction: from sklearn.decomposition import PCA

Common models & when to use:

- **Linear Regression** — regression with straight-line relationship.
- **Logistic Regression** — classification baseline; good when classes are linearly separable.
- **Decision Trees / Random Forests** — handle nonlinearities and mixed data without heavy preprocessing.
- **k-NN** — simple and intuitive; works well for small datasets and when distance is meaningful.
- **SVM** — strong for small to medium datasets with clear margins.
- **Neural networks** — when you have lots of data, especially for images/text.

Start with Linear Regression and then upgrade it to a better model*

-

4. **Evaluation Metrics**
    - **Sklearn.metrics (https://scikit-learn.org/stable/modules/model_evaluation.html)**

Examples: Regression, Classification, Classification