Different types of files

Fundamentally, all computer files are stored as binary data (sequences of 0s and 1s). However, we categorize them as "text" or "binary" based on **how that binary data is *interpreted***.

Here's a breakdown with examples:

1. Text Files

- **Interpretation:** The binary data inside a text file is interpreted as human-readable characters (letters, numbers, symbols). There's usually a specific character encoding (like UTF-8, ASCII, or ISO-8859-1) that dictates how these binary patterns map to characters.
- **Human Readability:** You can open a text file in a simple text editor (like Notepad, Sublime Text, VS Code) and read its contents directly.
- **Editing:** Easily editable with text editors.
- **Structure:** Often line-oriented, with newline characters (\n) separating lines of text.
- **Use Cases:**
  - **Source Code:** .py (Python), .java (Java), .html (Web pages), .css (Styling).
  - **Configuration Files:** .ini, .cfg, .xml, .json, .yaml. These are often structured text.
  - **Plain Text Documents:** .txt.
  - **Log Files:** Records of events or messages.
  - **Data Files:** .csv (Comma Separated Values) for tabular data.

2. Binary Files

- **Interpretation:** The binary data inside a binary file is *not* interpreted as human-readable characters. Instead, it's interpreted by specific programs or hardware according to a predefined format.
- **Human Readability:** If you open a binary file in a simple text editor, you'll likely see a jumble of seemingly random, unreadable characters,

symbols, or even blank spaces. This is because the text editor is trying to interpret non-character bytes as characters.

- **Editing:** Requires specialized software designed for that specific binary format. You can't just edit an image in Notepad.
- **Structure:** Often has a complex internal structure defined by its file format specification, which dictates where different pieces of data (e.g., pixel data, audio samples, executable instructions) are stored.
- **Use Cases:**
    - **Images:** .jpg, .png, .gif, .bmp. These files contain raw pixel data and metadata.
    - **Audio Files:** .mp3, .wav, .aac. Contain digital sound waves.
    - **Video Files:** .mp4, .avi, .mov. Contain compressed video and audio streams.
    - **Executable Programs:** .exe (Windows), .dmg (macOS), .bin (Linux/Unix binaries). These contain machine code instructions for the computer's processor.
    - **Compressed Archives:** .zip, .rar, .tar.gz. Contain compressed versions of other files.
    - **Microsoft Office Documents:** .docx, .xlsx, .pptx. While they can contain text, their underlying format is a complex binary structure (often zipped XML, but still binary at the lowest level).
    - **Databases:** .db, .sqlite. The internal storage format of a database.
- **Example:** You cannot directly "write" an example for a binary file in text here, as its contents would be unreadable. But imagine opening a .jpg image file in Notepad; you would see gibberish characters. That gibberish is the text editor's failed attempt to interpret the binary pixel data as text characters.

3. Below are common file formats used data science/ML, tagged as text or binary:

- **CSV (.csv):** Text
- **JSON (.json):** Text
- **TXT (.txt):** Text
- **Parquet (.parquet):** Binary
- **ORC (.orc):** Binary
- **Feather (.feather):** Binary
- **HDF5 (.h5, .hdf5):** Binary
- **Pickle (.pkl, .pickle):** Binary
- **NumPy (.npy, .npz):** Binary
- **Protobuf (.proto):** Binary (often serialized to binary format)
- **SQL (database dumps/scripts):** Text (the script itself) / Binary (the database file itself, e.g., SQLite .db) - *often treated as a data source rather than a file format in this context*
- **XLSX (.xlsx):** Binary (structured as zipped XML, but fundamentally binary)
- **Image formats (.png, .jpg, .jpeg, .gif, .tiff):** Binary
- **Audio formats (.mp3, .wav):** Binary