**Drug Feature Notes**

**🧩 Part 1: CSV Files and Dataset Overview**

* **CSV Files Used**:
  1. main\_database\_with\_properties.csv: Core drug information with physicochemical and pharmacological properties.
  2. drug\_interactions.csv: Drug-to-drug interaction details.
  3. drug\_categories.csv: Contains drug categories and MeSH IDs.
  4. experimental\_properties.csv: Feature-rich chemical properties per drug.
  5. pathways.csv: Contains pathway names, enzymes, categories, etc.
* **Discussion Points**:
  1. Drugs have incomplete features.
  2. Drugs and interactions are linked by primary\_drugbank\_id.
  3. Features like molecular\_weight, Boiling Point, pKa, and SMILES need cleaning.
  4. Need to structure everything for both ML (numerical features) and DL (encoded representations).

**🧪 Part 2: Feature Cleaning and Parsing Functions**

**1. extract\_boiling\_point(text)**

* **Purpose**: Parses messy boiling point strings like 78 °C, >600 °C, 398ºC (estimate) and extracts a usable float.
* **Logic**:
  + Finds float or range values.
  + If range, calculates the average.
  + Strips units like °C, mmHg.
  + Handles values like > 300, Decomposes as NaN.

**2. extract\_melting\_point(text)**

* **Purpose**: Cleans melting point text like 85.5-86.5 °C, about 190 and 245-247 °C, > 300 °C.
* **Logic**:
  + Extracts multiple ranges and averages them.
  + Normalizes numeric values.

**3. extract\_isoelectric\_point(text)**

* **Purpose**: Extracts numeric isoelectric points from strings like 5.35-6.20, ignoring notes like It does not possess a distinct value.
* **Logic**:
  + Extracts numbers and returns average if a range is given.

**4. extract\_float(text)**

* **Purpose**: Used to convert entries like 4.64e-02 g/l, 100 mg/ml, >10 mg/ml into g/L.
* **Logic**:
  + Converts units to grams.
  + Handles scientific notation.

**5. extract\_pka(text)**

* **Purpose**: Handles values like 9.62 (acid), 5.4 and 7.2, 15.06 (Strongest acidic).
* **Logic**:
  + Extracts all float-like numbers.
  + Computes average.

**🧬 Part 3: SMILES Processing**

**1. encode\_smiles(smile)**

* **Purpose**: Encodes a SMILES (chemical string) into a consistent integer hash for ML compatibility.
* **Logic**:
  + Uses SHA-256 hashing.
  + Reduces hash to 8 digits via modulus.

**2. extract\_smiles\_features(smile)**

* **Purpose**: Extracts structural features from the SMILES string like:
  + Length
  + Count of atoms: C, N, O, S, P
  + Count of bonds: =, #, @

**Reason for Two Approaches:**

* **Hashed SMILES**: Good for classical ML as categorical representation.
* **SMILES Features**: Better for analysis and interpretable modeling.

**🔗 Part 4: Interaction Dataset Label Engineering**

**Problem:**

* drug\_interactions.csv had only text descriptions like:
  + “The risk or severity of bleeding and bruising can be increased when Lepirudin is combined with Apixaban.”

**Goal:**

* Transform these into structured **interaction type** and **labels**.

**1. simplify\_interaction\_desc(desc)**

* **Purpose**: Extracts generalized interaction type from description.
* **Logic**:
  + Removes drug names.
  + Extracts common templates.

**2. interaction\_type\_category(desc)**

* **Purpose**: Labels interaction into one of several buckets (e.g. increases\_efficacy, bleeding\_risk, antagonistic).
* **Logic**:
  + Searches for keywords like increase, decrease, efficacy, toxicity, bleeding, etc.

**Labeling Output:**

* interaction\_label: integer label for classification.
* interaction\_type: simplified type string.

**📁 Part 5: Pathway Integration**

**Discussion:**

* Pathway data has multiple entries per drug.
* Grouping them makes the main dataset larger but informative.

**Logic:**

* Grouped by primary\_drugbank\_id.
* Aggregated:
  + Count of pathways.
  + Unique enzyme count.
  + Flag whether pathway info exists.

**Result:**

* Merged into main\_database\_with\_pathways.csv

**🧮 Part 6: Dataset Structuring for Machine Learning**

**Goals:**

* Build a supervised ML model to predict drug interactions.
* Explore individual and ensemble models (e.g. SVM, RF, LR, DT).

**Dataset Linking Strategy:**

* main\_database\_cleaned\_and\_encoded.csv (drug features).
* drug\_interactions\_encoded.csv (interaction pairs and labels).

**Merge Plan:**

* Merge drug A and drug B info using their respective drugbank IDs into interaction dataset.
* Create ML training data with feature combinations: Drug A + Drug B + interaction label.

**Primary Feature Engineering Steps:**

* Cleaned incomplete rows, missing data.
* Merged pathway info.
* Decoded/structured interaction descriptions.
* Generated new numerical/structural features.

**⚡ Part 7: Toxicity Feature Engineering — Attempts, Challenges & Next Steps**

**Summary of What We Tried:**

* **Parsed numeric ranges like 0.1-1000 mg/kg, computed average toxicity values.**
* **Extracted keywords: LD50, overdose, safe, mutagenic, rare, etc.**
* **Attempted to flag species-specific toxicity info (e.g., in rats, monkeys, humans).**

**Why It Wasn't Satisfying:**

* **No consistency across entries — each drug had unstructured free-text.**
* **Many valid values were nested in parenthetical or narrative context.**
* **Parsing missed context like which species a value belonged to.**
* **Too many assumptions about which value is "the" correct toxicity.**

**What We Can Do (Suggestions):**

* **Structured Value Column:**
  + **Extract and store all mg/kg numeric ranges by species as raw strings.**
  + **Example: toxicity\_ranges\_raw = "mouse: 1-1250; rat: 1-500"**
* **Average by Species:**
  + **Calculate avg toxicity per species and include columns:**
    - **toxicity\_mouse\_avg, toxicity\_rat\_avg, etc.**
* **Flag Drug Toxicity Mentions:**
  + **Add binary flags:**
    - **mentions\_safe\_terms, mentions\_overdose, mentions\_ld50, mentions\_rare\_effects, mentions\_unknown\_info**
* **Toxicity Severity Score *(optional)*:**
  + **Assign heuristic score based on combination of toxic keywords, presence of numeric values, and species data.**
* **Token Frequency Vectorizer *(advanced)*:**
  + **Use NLP to extract most common toxicity descriptions and categorize them.**

**🔢 Part 8: Preparing for Model Training**

**Step 1: Finalize Dataset**

* **Use drug\_interactions\_encoded.csv as label source.**
* **For each row:**
  + **Join drug A and drug B data from main\_database\_cleaned\_and\_encoded.csv.**
* **Form complete training table: [features of drug A] + [features of drug B] + interaction\_label**

**Step 2: Feature Selection**

* **Drop identifiers: primary\_drugbank\_id, name**
* **Remove high-missing-value columns or impute values**
* **Keep:**
  + **Cleaned numerics: Boiling Point, pKa, MW**
  + **SMILES features: carbon\_count, double\_bond\_count, etc.**
  + **Bioavailability, Rotatable Bond Count, logP, logS**
  + **Pathway counts**

**Step 3: Train ML Models**

* **Logistic Regression**
* **Support Vector Machine (SVM)**
* **Decision Tree**
* **Random Forest**

**Step 4: Evaluate Each**

* **Use cross-validation**
* **Track: accuracy, F1-score, precision, recall**

**Step 5: Build Weighted Ensemble**

* **Use soft voting or stacking**
* **Assign weights based on:**
  + **Each model's F1-score or AUC**
  + **Example: Final\_Pred = 0.1\*LR + 0.4\*RF + 0.2\*SVM + 0.3\*DT**