Let's go through each file one by one, explaining what it contains, how it is prepared, and how it is used in your model or code.

**1. ATCSimilarityMat.csv**

**Contents:**

* This file contains a similarity matrix based on the **Anatomical Therapeutic Chemical (ATC) classification system**.
* The ATC system classifies drugs based on their therapeutic properties and mechanism of action.
* Each row and column represent a drug, and the values indicate the similarity between two drugs based on ATC codes.

**Preparation:**

* The ATC codes for each drug are obtained from a drug database.
* A similarity score is calculated between each pair of drugs based on how closely their ATC codes match.
* The result is stored as a matrix where each cell (i, j) represents the ATC similarity between **drug i** and **drug j**.

**Usage in Model:**

* This similarity matrix is loaded into the model and used to extract **ATC-based features** for each drug pair.
* These features help the model learn interactions between drugs based on their therapeutic use.

**2. chem\_Jacarrd\_sim.csv**

**Contents:**

* This file contains a **chemical structure similarity matrix** based on the **Jaccard similarity index**.
* The Jaccard similarity is used to compare the chemical structures of two drugs.

**Preparation:**

* Drug structures are represented as molecular fingerprints.
* Jaccard similarity is calculated for each pair of drug fingerprints.
* The similarity values are stored in a matrix.

**Usage in Model:**

* The model uses this similarity matrix to understand how chemically similar two drugs are.
* If two drugs have similar chemical structures, they might interact in a similar way in the body.

**3. chemicalSimilarityMat.csv**

**Contents:**

* Another similarity matrix representing **chemical properties** of drugs.
* This may be computed using different methods such as **Tanimoto similarity, Euclidean distance, or cosine similarity**.

**Preparation:**

* Chemical properties (such as molecular weight, hydrogen bond donors/acceptors, logP, etc.) are extracted for each drug.
* A similarity score is computed based on these properties.

**Usage in Model:**

* This feature helps in predicting drug interactions based on **molecular properties**.
* If two drugs share many chemical properties, they might interact in a similar way.

**4. ddiMatrix.csv (Drug-Drug Interaction Matrix)**

**Contents:**

* This matrix contains known interactions between drug pairs.
* Each row and column represent a drug, and a **1 (interaction)** or **0 (no interaction)** indicates whether a known interaction exists.

**Preparation:**

* Extracted from a known drug-drug interaction database such as **DrugBank, KEGG, or ChEMBL**.
* If two drugs have reported interactions, the corresponding matrix cell is set to **1**.

**Usage in Model:**

* Used as **ground truth labels** during training.
* Helps the model learn to predict whether two drugs will interact based on other similarity features.

**5. drug\_drug\_matrix.csv**

**Contents:**

* Similar to ddiMatrix.csv, it represents drug-drug interactions.
* However, it might store interactions in a **different format (e.g., list form instead of a matrix).**

**Preparation:**

* Extracted from biomedical research databases.
* Processed into a structured format suitable for machine learning models.

**Usage in Model:**

* Used as input data to train and evaluate drug interaction predictions.

**6. drug\_fea.npy**

**Contents:**

* A NumPy file (.npy format) that contains precomputed **feature vectors** for each drug.
* Each row represents a drug, and columns contain numerical values representing different features.

**Preparation:**

* Features are extracted from multiple sources such as:
  + Chemical properties
  + ATC classification
  + Side effects
  + Molecular fingerprints
* These features are stored in an efficient NumPy array.

**Usage in Model:**

* Used as input to the deep learning model.
* Instead of recalculating features every time, the model loads this precomputed feature matrix for fast training and inference.

**7. drug\_side\_effects\_targets.csv**

**Contents:**

* Contains **side effects** and **target sites** for each drug.
* Columns:
  + **Drug ID**
  + **Drug Name**
  + **Side Effects**
  + **Target Sites** (biological targets like enzymes, receptors)

**Preparation:**

* Extracted from side effect databases such as **SIDER**.
* Matched with the drug database.

**Usage in Model:**

* Helps the model learn if side effects or target sites correlate with drug interactions.
* Used for displaying information in the **Streamlit app**.

**8. enzyme\_Jacarrd\_sim.csv**

**Contents:**

* A similarity matrix based on **enzyme targets**.
* Some drugs target the same enzymes, leading to potential interactions.

**Preparation:**

* Extracted from **biological interaction databases**.
* Jaccard similarity is calculated for drugs targeting the same enzymes.

**Usage in Model:**

* Helps in predicting interactions based on **biological mechanisms**.

**9. final\_model.keras**

**Contents:**

* This is the **trained deep learning model** saved in **Keras format**.
* It contains:
  + Neural network architecture
  + Model weights
  + Optimized parameters

**Preparation:**

* The model was trained using:
  + **Drug similarity matrices**
  + **Side effects data**
  + **Drug interaction labels**
* It learns patterns and predicts whether two drugs will interact.

**Usage in Model:**

* This model is **loaded in Streamlit** for real-time predictions.
* Given two drugs, it outputs:
  + Whether an interaction exists.
  + Probability of interaction.

**10. bg.jpg**

**Contents:**

* An image file used as a **background** in the Streamlit web application.

**Usage in Model:**

* It is **not used in the ML model** but **enhances UI/UX** in the web app.

**Summary Table:**

| **File Name** | **Contents** | **Usage in Model** |
| --- | --- | --- |
| ATCSimilarityMat.csv | ATC-based drug similarity | Feature input |
| chem\_Jacarrd\_sim.csv | Chemical structure similarity | Feature input |
| chemicalSimilarityMat.csv | Chemical property similarity | Feature input |
| ddiMatrix.csv | Known drug interactions | Ground truth labels |
| drug\_drug\_matrix.csv | Drug interactions in different format | Ground truth labels |
| drug\_fea.npy | Drug feature vectors (precomputed) | Model input |
| drug\_side\_effects\_targets.csv | Side effects & target sites | Additional features for analysis |
| enzyme\_Jacarrd\_sim.csv | Enzyme-based drug similarity | Feature input |
| final\_model.keras | Trained deep learning model | Model inference |
| bg.jpg | Background image for UI | Web app design |

**Next Steps**

Now that we understand the dataset, you can:

1. **Load these files in Jupyter Notebook** to visualize them.
2. **Train or retrain the model using these datasets**.
3. **Run the Streamlit app to use the trained model for predictions**.

Would you like a **step-by-step guide on running the model in Jupyter Notebook**? 😊

I'll analyze each of your uploaded datasets and explain them one by one, including their contents, preparation, and usage in your model. Let’s begin.

**1. fused\_similarity.npy**

**Contents:**

* This file contains a **precomputed similarity matrix** combining multiple similarity measures.
* It likely integrates similarities from different sources (e.g., chemical structure, ATC classification, side effects, etc.).

**Preparation:**

* Different similarity matrices (chemical, target-based, pathway-based, side effect-based) are computed separately.
* These matrices are fused using a weighted approach or another fusion method (e.g., matrix factorization or averaging).
* The final combined matrix is stored in NumPy format (.npy) for efficient loading.

**Usage in Model:**

* Used as input features to predict drug interactions.
* Instead of using multiple similarity matrices separately, the model can directly use this **fused** representation.

**2. offsideeffect\_Jacarrd\_sim.csv**

**Contents:**

* This file contains a similarity matrix based on **Jaccard similarity of side effects** between drugs.
* Each row and column represent a drug, and the values indicate the **degree of similarity** based on shared side effects.

**Preparation:**

* Side effect data is obtained from sources like **SIDER**.
* Each drug’s side effects are represented as a binary vector (1 if the drug has a side effect, 0 otherwise).
* Jaccard similarity is computed between drug pairs.

**Usage in Model:**

* Helps predict interactions by understanding how **shared side effects** influence drug interactions.
* Drugs with similar side effects may have overlapping mechanisms of action.

**3. pathway\_Jacarrd\_sim.csv**

**Contents:**

* A similarity matrix representing **pathway-based similarity** between drugs.
* Pathways refer to **biological routes** that drugs influence (e.g., metabolic pathways).

**Preparation:**

* Pathway data is collected from **KEGG or Reactome** databases.
* Each drug is mapped to the pathways it affects.
* Jaccard similarity is computed to compare pathway overlap between drug pairs.

**Usage in Model:**

* Helps the model understand if two drugs interact by **targeting the same pathways**.
* Useful for predicting interactions based on shared mechanisms.

**4. SideEffectSimilarityMat.csv**

**Contents:**

* This file contains a **side effect similarity matrix**.
* Similar to offsideeffect\_Jacarrd\_sim.csv, but might be computed using a **different method** (e.g., cosine similarity instead of Jaccard).

**Preparation:**

* Side effect profiles of drugs are extracted from **SIDER**.
* A similarity score is calculated between each pair of drugs based on side effect overlap.

**Usage in Model:**

* Used to train the model to recognize potential interactions based on **side effect similarities**.
* If two drugs cause similar adverse reactions, they might interact in a specific way.

**5. similarity\_matrices.npy**

**Contents:**

* A NumPy array containing **multiple similarity matrices**.
* Instead of storing them separately, all similarity matrices are saved in a structured .npy file.

**Preparation:**

* Individual similarity matrices (chemical, ATC, pathway, enzyme, target, side effect) are computed.
* These matrices are stored in a single file for fast loading.

**Usage in Model:**

* The model loads this file to access all similarity matrices at once.
* Used for feature extraction and representation learning.

**6. target\_Jacarrd\_sim.csv**

**Contents:**

* A similarity matrix based on **biological targets** of drugs.
* Each value represents how similar two drugs are based on **their interaction with proteins, enzymes, or receptors**.

**Preparation:**

* Drug-target interaction data is gathered from **ChEMBL, DrugBank, or KEGG**.
* Each drug is associated with proteins or enzymes it binds to.
* Jaccard similarity is computed between drugs based on shared targets.

**Usage in Model:**

* Helps predict interactions based on **biological function similarity**.
* If two drugs target the same proteins, they might have **synergistic or antagonistic interactions**.

**Summary Table:**

| **File Name** | **Contents** | **Usage in Model** |
| --- | --- | --- |
| fused\_similarity.npy | Combined similarity matrix | Feature input |
| offsideeffect\_Jacarrd\_sim.csv | Side effect similarity (Jaccard) | Feature input |
| pathway\_Jacarrd\_sim.csv | Pathway-based similarity | Feature input |
| SideEffectSimilarityMat.csv | Side effect similarity matrix | Feature input |
| similarity\_matrices.npy | Multiple similarity matrices | Feature input |
| target\_Jacarrd\_sim.csv | Drug-target similarity (Jaccard) | Feature input |

**Next Steps**

* Would you like to **load and visualize** these files in Python to better understand their content?
* Do you need help with **training or debugging your model**? 😊