**What This Code Does:**

1. **Imports required libraries**:
   * rdkit for chemical descriptor calculations
   * torch for deep learning
   * matplotlib, numpy, pandas for data handling and visualization
2. **Defines chemical components**:
   * Uses **SMILES** strings for Pigment (TiO₂), Extender (CaCO₃), and Binder (Polyvinyl Acetate).
   * Computes **molecular descriptors** (Molecular Weight, TPSA, LogP, Heavy Atom Count) using RDKit.
3. **Loads experimental dataset**:
   * Contains **Pigment, Extender, Binder amounts** and their **Viscosity, Density, and pH** as target outputs.
4. **Feature Engineering**:
   * Generates **new chemical features** by combining component properties with their weights.
   * Includes **cross-component interactions** (e.g., Pigment \* Extender \* LogP).
5. **Data Preprocessing**:
   * Splits data into **training & test sets** (80-20).
   * Normalizes data using StandardScaler.
   * Converts data into PyTorch tensors.
6. **Builds a Neural Network**:
   * **4-layer architecture**:
     + Input → 32 neurons (ReLU) → 16 neurons (ReLU) → 8 neurons (ReLU) → Output (3 neurons for Viscosity, Density, pH).
   * Uses **Dropout (0.3) to prevent overfitting**.
   * Trained with **Adam optimizer & MSE Loss**.
   * Implements **early stopping** to prevent overfitting.
7. **Training the Model**:
   * Runs for **max 5000 epochs** but stops early if validation loss doesn’t improve for 100 epochs.
8. **Evaluation & Visualization**:
   * Plots **training vs validation loss**.
   * Compares **actual vs predicted values** for each property.
9. **Feature Importance Analysis**:
   * Uses **permutation importance** to check which features impact predictions most.
   * Plots **feature importance scores**.
10. **Predicts Formulation Properties for New Data**:

* Loads **new formulation dataset**.
* Predicts **Viscosity, Density, pH** using the trained model.
* Displays the predicted values in a formatted table.