EnhancedDiseaseSearch Model - Detailed Explanation

# Overview

The EnhancedDiseaseSearch model processes user symptoms and metadata (age, gender, pregnancy status) to return a ranked list of likely diseases along with relevant treatment and precautionary information. It uses a hybrid of:  
- TF-IDF + Cosine Similarity for textual symptom similarity  
- CountVectorizer + Multinomial Naive Bayes for probabilistic disease prediction  
- Filtering and score adjustment based on patient context

# 1. Data Preprocessing

The `load\_data()` function loads a CSV file `augmented\_diseases\_extended.csv`, merging various columns into aggregated fields:  
- `Symptoms`: Merged from Symptom1-Symptom4  
- `Protocol`, `Comorbidities`, `Precautions`, `Diet`, `Workout`: Combined and cleaned  
  
These fields are used downstream for matching and recommendation.

# 2. Model Initialization

The `\_\_init\_\_` method performs two key tasks:  
- TF-IDF Vectorization: Transforms symptom text into vectors for similarity calculations.  
- Naive Bayes Training: Uses CountVectorizer and trains a Multinomial Naive Bayes model for symptom → disease prediction.

# 3. Symptom Matching with TF-IDF (search method)

The `search()` method is the core function for retrieving disease matches:  
1. Normalize input symptoms and convert to vector using TF-IDF.  
2. Compute cosine similarity with disease symptom vectors.  
3. Apply demographic filters (age, gender, pregnancy safety).  
4. Count exact symptom matches.  
5. Adjust match scores based on:  
 - Symptom overlap  
 - Age/gender match  
 - Pregnancy safety  
6. Return top N diseases sorted by adjusted similarity scores, with detailed metadata.

# 4. Prediction with Naive Bayes (predict\_with\_naive\_bayes)

Uses the trained Naive Bayes model to predict the most likely diseases based on symptom text.  
- Input: List of symptoms  
- Output: Top N diseases with associated prediction probabilities.

# 5. Helper Functions

- `\_check\_age\_match()`: Matches age to age group tags like 'pediatric', 'adult', etc.  
- `\_check\_gender\_match()`: Validates if disease applies to the user’s gender.  
- `\_check\_pregnancy\_safety()`: Penalizes diseases that are unsafe during pregnancy or planning.

# Summary Flow

1. User inputs symptoms, age, gender, pregnancy status.  
2. System computes similarity using TF-IDF.  
3. Filters out incompatible diseases.  
4. Adjusts scores using demographic and symptom match heuristics.  
5. Returns detailed top-N matching diseases.  
6. Uses Naive Bayes to suggest diseases probabilistically.