**AI-Based Wellness Scoring System Framework**

**1. Data Collection and Preprocessing**

* **Objective: Gather and prepare data from multiple sources (Health, Life Events, Social Trends, Urban Issues) to ensure consistent, normalized inputs for the scoring model.**
* **Steps:**
  + **Aggregate Data: Retrieve individual data points from the Health Database (e.g., heart rate, sleep quality), Life Event Database (e.g., job change, health incidents), Family Score Database, and Community Score Database.**
  + **Data Cleaning: Remove or handle missing data, address outliers, and correct any inconsistencies.**
  + **Normalization: Standardize continuous variables such as health metrics (heart rate, activity levels) and event frequencies to a common scale (e.g., 0–1 or mean of 0 and variance of 1).**
    - ***Python Packages*: pandas, sklearn.preprocessing (e.g., StandardScaler, MinMaxScaler)**
  + **Categorical Encoding: Convert categorical data (e.g., event types like ‘health’ or ‘economic’) into numerical representations through one-hot encoding or embeddings.**
    - ***Python Packages*: sklearn.preprocessing (e.g., OneHotEncoder)**

**2. Weighted Scoring Calculation**

* **Objective: Apply weights to different types of inputs (health, life events, social trends) to calculate an initial wellness score.**
* **Steps:**
  + **Define Weights: Set predefined weights for each data category based on its significance. For example:**
    - **Health data might have a higher weight (e.g., 0.6) since it directly impacts individual wellness.**
    - **Life events (e.g., job changes, personal milestones) could have a medium weight (e.g., 0.3).**
    - **Social trends and urban issues might have a lower weight (e.g., 0.1) if they represent external influences.**
  + **Calculate Individual Scores: Multiply each input by its assigned weight. For example, if health\_score = 0.8, life\_event\_score = 0.5, and social\_trend\_score = 0.3, the individual score contributions would be: Weighted Health Score=0.8×0.6=0.48\text{Weighted Health Score} = 0.8 \times 0.6 = 0.48Weighted Health Score=0.8×0.6=0.48 Weighted Life Event Score=0.5×0.3=0.15\text{Weighted Life Event Score} = 0.5 \times 0.3 = 0.15Weighted Life Event Score=0.5×0.3=0.15 Weighted Social Trend Score=0.3×0.1=0.03\text{Weighted Social Trend Score} = 0.3 \times 0.1 = 0.03Weighted Social Trend Score=0.3×0.1=0.03**
  + **Aggregate Weighted Scores: Sum the weighted scores to obtain the initial wellness score. Initial Wellness Score=0.48+0.15+0.03=0.66\text{Initial Wellness Score} = 0.48 + 0.15 + 0.03 = 0.66Initial Wellness Score=0.48+0.15+0.03=0.66**
    - ***Python Packages*: numpy for vectorized operations**

**3. Machine Learning for Predictive Scoring**

* **Objective: Use a regression model to refine the wellness score by accounting for correlations between past life events and health metrics, as well as predicting future impacts.**
* **Steps:**
  + **Model Training: Train a regression model (e.g., linear regression, decision tree, or gradient boosting) on historical data to predict the wellness score based on current inputs.**
    - **Input Features: Health metrics, life event frequency/severity, social trend data.**
    - **Target: Wellness score from historical records.**
    - ***Python Packages*: sklearn.linear\_model for linear regression, xgboost for gradient boosting**
  + **Predictive Adjustment: Once the model is trained, use it to predict the impact of new data points on the wellness score. For example, if a user logs a major health event, the model predicts how this will influence their future wellness score.**
  + **Combine Predictions with Weighted Scores: Adjust the initial wellness score by averaging it with the predictive model’s output to refine the score: Refined Wellness Score=Initial Wellness Score+Predicted Score2\text{Refined Wellness Score} = \frac{\text{Initial Wellness Score} + \text{Predicted Score}}{2}Refined Wellness Score=2Initial Wellness Score+Predicted Score​**

**4. Family and Community-Level Adjustments**

* **Objective: Aggregate individual wellness scores to produce family and community-level scores, adjusting based on broader trends.**
* **Steps:**
  + **Family Score Calculation: Average the wellness scores of all individuals in a family. Adjust this score based on collective family events or shared health data (e.g., if a family member experiences a severe health incident, it might slightly lower the family score).**
    - ***Formula*: Family Score=∑Individual Wellness ScoresNumber of Family Members\text{Family Score} = \frac{\sum \text{Individual Wellness Scores}}{\text{Number of Family Members}}Family Score=Number of Family Members∑Individual Wellness Scores​**
  + **Community Score Calculation: Aggregate family scores across the community and incorporate social trends. For instance, if the community is experiencing a high rate of unemployment, the community score might be adjusted downward.**
    - ***Formula*: Community Score=∑Family ScoresNumber of Families+Adjustment Factor (e.g., unemployment rate)\text{Community Score} = \frac{\sum \text{Family Scores}}{\text{Number of Families}} + \text{Adjustment Factor (e.g., unemployment rate)}Community Score=Number of Families∑Family Scores​+Adjustment Factor (e.g., unemployment rate)**
  + ***Python Packages*: numpy for aggregation, pandas for data manipulation**

**5. Continuous Learning and Adaptation**

* **Objective: Implement a feedback loop that adjusts the scoring weights and model parameters over time, making the system responsive to new data patterns.**
* **Steps:**
  + **Retraining the Model: Periodically retrain the model using new data to ensure the predictive scoring remains accurate as social and personal dynamics change. This can be done daily, weekly, or monthly, depending on the data update frequency.**
    - ***Python Packages*: sklearn, tensorflow or pytorch for neural network-based models if more sophisticated learning is desired.**
  + **Adaptive Weight Adjustment: Use recent performance data to adjust the weights assigned to different categories dynamically. For example, if health data consistently has a stronger correlation with overall wellness, the system might increase its weight over time.**
  + **User Feedback Integration: Incorporate feedback from users or other stakeholders to refine scoring mechanisms. For instance, users could rate the relevance of their wellness score, and this feedback could guide future model updates.**

**6. Score Output and Reporting**

* **Objective: Output the final scores for individuals, families, and communities, providing a clear, interpretable wellness score that reflects recent data.**
* **Steps:**
  + **Store the Calculated Scores: Write the final scores to the appropriate databases: User Score Database for individual scores, Family Score Database for family-level scores, and Community Score Database for aggregated community wellness.**
  + **Generate Reports: Prepare visualizations and summaries of the wellness scores over time, enabling users and city officials to track trends and identify areas for improvement.**
    - ***Python Packages*: pandas for data storage, matplotlib or seaborn for visualization**