**LOGIC OF THE CODE**

The core of this project is a recommendation engine designed to intelligently match Founders with relevant Service Providers and Mentors. The matching process is powered by a weighted scoring algorithm that calculates a comprehensive "match score" out of 100 for each potential pairing.

**1. Data Loading and Preparation**

First, the code loads a dataset from a CSV file using the pandas library. It then splits this data into two distinct groups:

* **founders\_df**: A table containing only founders (identified by a user\_id starting with 'F').
* **providers\_df**: A table containing only service providers and mentors (identified by a user\_id starting with 'S').

**2. The Scoring System: Three Key Components**

The core of the logic lies around a scoring system that evaluates a match based on three criteria. A final weighted score is calculated from these components.

**A. Semantic Skill & Need Matching (50% Weight)**

Instead of simply matching keywords (like "Python" with "Python"), it uses a pre-trained AI model (all-MiniLM-L6-v2 from the sentence-transformers library) to understand the meaning behind the text.

The logic is as follows:

1. **Text to Vectors**: The model converts a founder's need (e.g., "I need a new mobile app") and a provider's skill (e.g., "Expert in full-stack development") into numerical vectors or embeddings.
2. **Cosine Similarity**: It then calculates the **cosine similarity** between these vectors. This produces a score indicating how semantically close the two phrases are. For example, "UI/UX revamp" and "Design expert" are understood to be related, even though they don't share many keywords.
3. **Final Score**: This similarity value is converted to a score from 0 to 100.

This is done for both tech\_requirement vs. core\_skill and project\_need vs. expertise\_area, and the two scores are averaged.

**B. Industry Match (30% Weight)**

It checks if the provider's industry preference aligns with the founder's industry.

* **Perfect Match (100 points)**: Founder's industry (e.g., "FinTech") is the same as the provider's preference ("FinTech").
* **Good Match (70 points)**: The provider is open to working in 'Any' industry.
* **Mismatch (0 points)**: The industries do not align.

**C. Timeline & Availability Match (20% Weight)**

This component ensures that the provider is available when the founder needs them. It works by mapping text-based deadlines and availabilities to numerical values for comparison.

* **Match (100 points)**: The provider is available at or before the founder's project deadline.
* **Mismatch (0 points)**: The provider is not available on time.

**3. Calculating and Presenting Matches**

With the scoring logic in place, the code performs the following steps:

1. **Iterate and Score**: It loops through every founder and every provider, calculating the final weighted match score for each possible pair using the calculate\_match\_score function.
2. **Generate Recommendations**: It creates two functions:
   * get\_top\_providers\_for\_founder(): Finds the best provider matches for a given founder.
   * get\_top\_founders\_for\_provider(): Finds the best project matches for a given provider.
3. **Summarize and Explain**: The final output generates two clean tables: one with recommendations for a sample of founders and another for a sample of providers.

**I**t includes a "Reason for Match" column that explains why the recommendation was made, using the individual component scores.

1. **Visualize with a Heatmap**: To give a full overview of all possible matches, the code generates a heatmap. In this chart, each cell represents a founder-provider pair, and the color intensity shows the strength of their match score, making it easy to spot the strongest connections visually.

Lastly, I have created a Streamlit dashboard for better visualisation of the Recommendation Engine.

**User Role Toggle (Founder vs. Service Provider)**

* In the **sidebar**, users can choose:
* To view as a **Founder** or a **Service Provider**
* Then select a user ID from that category.

**User Profile Display**

* Once a user is selected, their **complete profile details** (as JSON) are displayed in the main area. It gives context to the match recommendations.

**Top 3 Recommendations**

* For the selected user, it shows their **Top 3 matched counterparts** based on the final match score, presented in a clean, scrollable st.dataframe() :
  + For Founders → Top 3 matching Service Providers
  + For Providers → Top 3 matching Founders

**Match Matrix Heatmap**

* Displays a **heatmap (matrix)**, built using Plotly Express for interactivity, showing match scores between:
  + All Founders (rows) and all Service Providers (columns)

**CSV Export of All Match Scores**

* A **download button** allows users to export the **full match results** as a .csv file.