



# 基於 PyTorch 的股票與基金個人化推薦系統實作指南

## 系統目標與特性

本指南旨在設計並實作一個個人化的投資推薦系統，可根據用戶特徵與市場數據，推薦符合用戶偏好和風險屬性的股票或基金產品。系統特色包括：支援台股與美股市場的推薦、可透過模式切換在「股票推薦」與「基金推薦」間自由切換，以及應對新用戶的冷啟動問題並提供可解釋的推薦結果。系統的目標是在兼顧用戶個人偏好與投資組合績效的前提下，提供精準且有說服力的投資建議<sup>1</sup>。過去研究指出，一個有效的股票推薦系統需要同時考慮個人偏好、投資組合分散化以及時間動態等要素<sup>1</sup>。此外，僅基於歷史績效進行推薦往往無法滿足用戶口味<sup>2</sup>；因此本系統透過整合豐富的用戶屬性來改善冷啟動情況，例如利用風險承受度等特徵來提升推薦多樣性<sup>3</sup><sup>4</sup>。

**主要功能：**- **個性化資產推薦**：根據用戶的年齡、風險偏好、投資目標等特徵，從台股與美股市場中篩選出最符合其需求的股票或基金。  
- **推薦模式切換**：系統可在「股票模式」與「基金模式」間切換，根據用戶選擇決定推薦的資產類型。在內部實現上，兩種模式將共享模型的大部分架構，但在輸入特徵處理與輸出層上做區別，以適應不同資產類別。  
- **新用戶冷啟動**：針對缺乏歷史交易或點擊記錄的新用戶，系統利用用戶提供的靜態特徵（如人口統計、風險偏好、投資經驗）來產生初步推薦。同時，採用個性化屬性（例如性格或風險傾向）有助於緩解冷啟動問題，提高推薦多樣性<sup>4</sup>。  
- **推薦結果解釋**：每筆推薦將附帶理由說明，例如「此基金近一年報酬率在同類排名中表現前茅」或「該股票所屬產業符合您的關注領域，且財務指標穩健」。透過提供這類自然語言的解釋以及相關數據指標，提升模型透明度與用戶信任<sup>5</sup><sup>6</sup>。  
- **績效指標展示**：除了傳統推薦評估指標（Precision@K、NDCG@K、Hit Rate等），系統亦會呈現所推薦投資組合的金融績效指標，如年化報酬率與夏普比率，幫助用戶評估推薦的投資價值。  
- **即時互動調整**：系統支援用戶調整偏好條件（例如改變投資目標或風險等級）後即時更新推薦結果。模型可以根據新的輸入特徵重新計算用戶嵌入向量，並快速檢索新的最相關資產，以動態回應用戶需求變化。

## 資料來源與特徵說明

實現此推薦系統需要整合多源異質數據，包括用戶屬性資料、股票市場數據、基金資料，以及（可選）資產關聯的圖譜數據。以下對各類資料及其特徵進行說明：

### 用戶特徵資料

- **人口統計與基本資訊**：用戶年齡、收入水平（選填）等，可作為數值或類別特徵輸入模型。例如年齡可經過區間離散化或標準化後餵入，收入水平可作為類別特徵嵌入。
- **風險偏好**：用戶的風險承受度類型（保守型／穩健型／積極型）。這可作為一個重要的類別特徵，決定推薦資產的風險屬性範圍。例如對保守型用戶傾向推薦波動較低、評級較安全的資產。
- **投資目標**：如退休儲蓄、資本增長、短期投機等。這些目標可映射成模型的偏好向量的一部分，影響推薦結果（例如偏重長期穩定回報或短期高增長標的）。
- **投資經驗**：用戶的投資經驗水平（新手、一般、熟練）。經驗不足的用戶可能更需要分散投資或低風險產品；此特徵可引導模型對風險高的資產降低分數。
- **歷史行為記錄**：包括過去曾買入或點擊瀏覽的股票與基金清單（如有）。這些歷史持倉/基金紀錄可用來推斷用戶興趣領域（偏好產業、股票型或債券型基金偏好等）。對有歷史資料的用戶，可將過往互動資產嵌入序列模型或簡單聚合生成用戶偏好向量的一部分<sup>7</sup>。對沒有歷史的用戶（冷啟動），則依賴上述靜態屬性進行推薦<sup>4</sup>。整合人格特質或風險傾向等資訊，有助於提升新用戶推薦質量<sup>8</sup>。

## 股票市場資料

- **價格時間序列**：每支股票的歷史日K線數據，包括開盤價、最高價、最低價、收盤價、成交量等時間序列。這是反映股票走勢和波動性的關鍵資料，可用於模型的序列子網路中。例如可以取最近1年或數月的每日價格序列作為模型輸入，經由1D-CNN或Transformer等提取時序特徵，以捕捉趨勢和動能訊號<sup>9</sup>。
- **技術指標**：由價格序列計算的技術分析指標，如移動平均線(MA)、相對強弱指數(RSI)、移動平均趨勢(MACD)、布林帶等。這些指標可定期預計算作為股票的特徵向量的一部分，用於輔助模型了解當前股價所處的技術形態（超買超賣、趨勢方向等）。
- **基本面財務指標**：股票對應公司的財務健康度與估值指標，例如每股盈餘(EPS)、淨資產收益率(ROE)、本益比(P/E)、股息率等。這些屬性多為數值型特徵，可透過適當變換（對數、標準化等）後餵入模型的數值特徵通道<sup>7</sup>。基本面指標有助於模型評估股票的內在價值和品質。
- **分析師評級**：市場上對該股票的綜合評級（例如買進/中立/賣出）或目標價預測。這可作為一個額外的特徵向量（例如one-hot或數值映射：1-5分）供模型參考，反映市場專業觀點。
- **產業分類**：股票所屬的產業或板塊（如科技、金融、醫療等）。可作為類別型特徵嵌入，用於捕捉行業趨勢及分散風險考量。如果用戶對某些產業特別偏好或迴避，模型可以學習到相應權重影響。產業分類也會用於圖譜構建，連結同行業的股票節點。
- **市場整體因子（可選）**：如市場指數走勢、行業指數，以提供當前市場行情的背景特徵。

## 基金產品資料

- **基金類型**：基金的類別（股票型、債券型、混合型、貨幣型等）。此類別特徵有助於模型了解基金的大致風險收益範疇，例如股票型基金風險較高而債券型相對穩健，對應不同用戶風險檔次。
- **歷史淨值時間序列**：基金過往淨值(NAV)的時間序列資料，以及由此衍生的階段性報酬率（如近1年、近3年報酬率）。類似股票價格序列，可將基金淨值序列輸入序列模型，以提取基金績效走勢特徵<sup>9</sup>。近1年/3年的報酬率則可直接作為數值特徵，表示基金的歷史績效。
- **風險評級**：如晨星(Morningstar)風險等級或基金波動度指標（標準差、Beta值）。這些特徵反映基金的風險水準，可與用戶風險偏好匹配。例如對保守型用戶，模型應偏好推薦風險評級低的基金。
- **基金經理與公司**：基金經理的歷史業績、公司信譽等資訊（如經理從業年限、歷年平均績效）。這些可用來加強模型對基金的信心評估（可能以數值或類別特徵表示，例如星級評分或屬性標籤）。
- **同類排名**：基金在相同類型產品中的績效排名（例如收益排名前10%等）。此特徵可作為模型解釋的一部分：高排名意味著該基金近期表現優於大多數同業，可能成為推薦理由之一。
- **基金持倉（可選）**：基金持有的前幾大資產列表（股票、債券等）。這可被用於圖譜數據，將基金節點與其投資標的（特別是股票）節點相連。透過這些連結，模型可以學習基金與股票之間的關聯度，例如兩支不同基金可能因共享多支持股而在隱含風險上相關。

## 資產關聯圖譜資料（可選）

為了進一步提升推薦效果與解釋能力，可構建**金融知識圖譜**或**關聯圖**，將股票、基金及相關實體透過關係連結起來，供**圖神經網路(GNN)**模組學習。可能的圖譜設計包括：

- **股票-產業圖**：節點包含股票和產業兩類，連線表示某股票屬於某產業。透過圖卷積或圖注意力網路，能夠讓模型學到同行業股票之間的相關性，以及產業景氣對股票的影響。如果某用戶在某產業上有偏好，其嵌入向量與該產業相關的股票節點會更相近。
- **基金-持股圖**：節點為基金和股票，連線表示基金持有該股票（可加權連線代表持倉比例）。這有助於模型了解基金間的相似度（持有相似股票的基金可能性質相近）以及股票與基金的關聯。一些研究已證明運用圖注意力網路來捕捉股票之間或資產之間隱含關係能提升模型表現<sup>10</sup>。例如，GAT能讓模型聚焦於圖中最相關的關係，提取資產之間的重要關聯資訊<sup>10</sup>。
- **其他關係**：如股票之間的競爭/合作關係、公司高管關聯、宏觀經濟因素等，也可視數據情況納入圖譜。

**圖譜數據的作用**在於彌補傳統特徵的不足，將隱含知識融入資產表示。例如，透過GNN可以學到某基金因持有某股票而間接受益於該股票近期上漲的信息，或學到產業鏈上一組股票的表現聯動。先前有學者提出動態圖注意力模型來追蹤隨時間變化的公司關聯，增進股票推薦效果<sup>11</sup>。

## 模型架構與方法

整個推薦系統模型採用**雙塔深度學習架構（Two-Tower Model）**<sup>12</sup> 實現，以 PyTorch 為基礎構建。雙塔模型將用戶和資產（股票/基金）的表示學習分成兩個塔進行，最終通過嵌入向量的相似度來預測用戶對資產的偏好分數<sup>13</sup>。這種架構在大型推薦系統中被廣泛採用，因為它能有效融合豐富的用戶和物件特徵，同時保持高效的檢索速度<sup>12</sup>。

模型主要組成與流程如下：

- **用戶塔（User Tower）**：專門負責編碼用戶特徵。將上述用戶的各種屬性輸入到用戶塔的神經網路中，產生用戶的低維稠密向量表示  $u$ <sup>14</sup>。具體而言，用戶塔包含若干子模組：
  - 對於類別型特徵（如風險偏好、投資目標、經驗水平），使用**嵌入層**將每種取值映射為稠密向量，再將多個嵌入向量串接或經過MLP融合<sup>7</sup>。例如風險偏好3類可映射為長度\$D\$的embedding向量。
  - 對於數值型特徵（如年齡、收入），先進行適當的歸一化或分桶處理，再作為MLP的輸入節點或轉換為1維嵌入向量<sup>7</sup>。
  - **歷史行為序列**（如用戶過去投資/點贊的資產序列，若有）可經由**序列模型**提取特徵：例如應用RNN/LSTM、1D-CNN或Transformer編碼用戶的交互序列<sup>7</sup>。這將產生一個代表用戶長期偏好的序列向量。對短序列也可採用平均池化或最近幾個交互的embedding均值作為用戶興趣概括。
- 最後將所有來自用戶特徵的向量經過拼接，送入若干全連接層（MLP）進一步非線性變換，輸出固定長度的用戶**嵌入向量** \$u\$<sup>14</sup>。該向量在嵌入空間中表徵了用戶的整體投資偏好傾向。
- **資產塔（Item Tower）**：負責編碼資產（股票或基金）的各種特徵，輸出對應資產的向量表示 \$v\$<sup>15</sup>。由於股票與基金的特徵結構不同，資產塔會針對模式切換進行**條件化處理**：
  - **股票模式**：當推薦股票時，資產塔啟用「股票子網路」來處理股票特徵。首先將股票的類別特徵（如產業類別）經嵌入層轉為向量；數值特徵（如財務比率、技術指標）進行標準化後輸入MLP。股票的歷史價格序列則餵入**時間序列子網路**（可選擇Transformer、LSTM或一維CNN）以擷取近期趨勢模式<sup>16</sup>。例如，一個Transformer Encoder可用於提取股票過去N日價格的時序特徵，在金融序列預測中已證實能抓住關鍵的時序依賴<sup>16</sup>。如有使用圖譜資訊，則將股票節點的初始特徵通過**圖神經網路模組**與其關聯節點（產業、相關股票、基金持股等）進行圖卷積/注意力計算，獲得經關聯信息強化的股票表示向量<sup>17</sup>。例如利用Graph Attention Network（GAT）聚合同產業股票鄰居節點的訊息，以考慮板塊趨勢影響<sup>10</sup>。最後，將股票的各類特徵向量（基本面、技術面、圖譜等）合併經過MLP得到該股票的最終嵌入向量 \$v\_{\text{stock}}\$<sup>16</sup>。
  - **基金模式**：當推薦基金時，資產塔使用「基金子網路」。處理方式類似但針對基金特性調整：基金類別經嵌入層得到類別向量，基金的數值型特徵（近1年/3年收益、風險評級、排名等）直接餵入全連接層。基金淨值的歷史序列可經過一個共享參數的時序模型（架構上可與股票的時序模型類似但不共享權重）提取波動與趨勢特徵。若圖譜有定義基金與股票的關係，也可透過圖神經網路將所持股票的近期表現訊號傳遞到基金節點。例如基金連接的股票節點最近表現都很好，則圖模型會為基金節點帶來積極的鄰居影響，有研究利用知識圖譜嵌入來同時獲得基金推薦和解釋<sup>6</sup>。最終，基金子網路輸出基金的嵌入向量 \$v\_{\text{fund}}\$<sup>16</sup>。
  - **參數共享與差異**：模型會在**大部分層共享**兩種模式的設計哲學，例如用戶塔對應任意模式都通用，且股票/基金塔可以在某些通用特徵上共用表示空間。但基於輸入維度差異和預測目標不同，股票與基金塔**不會完全共用參數**。可採取的實作方式包括：在PyTorch中定義一個包含股票與基金兩套子網的類別，根據輸入 `asset_type` 選擇對應的子網計算；或者在最後輸出層設計兩個**head**，一個專門用於預測股票偏好分，一個用於基金偏好分，各自對應各自的訓練目標和loss權重。這樣確保模型能針對不同資產類型學到專門的特徵表示與偏好分佈，同時維持架構的一致性。
  - **嵌入空間打分**：經上述用戶塔和資產塔計算得到的向量 \$u\$ 和 \$v\$，會在公共的嵌入空間中進行匹配打分。常用的打分函數是**向量點積或餘弦相似度**<sup>13</sup>。本系統採用點積：\$Score(u, v) = u \cdot v\$。其值代表用戶與該資產的親和程度。點積高表示用戶偏好強烈匹配該資產，低則表示不感興趣<sup>13</sup>。在模

型的訓練過程中，我們希望正樣本（用戶喜歡的資產）對應的 $u \cdot v$ 更大，負樣本（不感興趣的資產）點積較小。透過雙塔架構，推薦系統在服務時可以高效地執行向量檢索：對於給定用戶即時算出的 $u$ ，與預先計算並存儲好的所有資產嵌入集合 $\{v\}$ 進行近似最近鄰搜尋，以找出點積最高的Top-K項目<sup>18</sup>。這樣的兩階段檢索（先離線計算物件向量、在線ANN搜索）能滿足大型商品庫的毫秒級推薦需求<sup>19</sup>。

**架構拓展：**在雙塔的框架下，我們可以靈活引入新的模型元件：  
 - 使用**多頭注意力機制**改善特徵交互：例如在將用戶多個特徵融合時引入Attention，以關注最重要的用戶偏好因素；或在資產塔中對接不同模態（時序、基本面、圖譜）特徵時，引入注意力來自適應權重。  
 - 使用**多任務學習**：除了主任務（預測用戶-資產偏好），可增加輔助任務如預測資產未來收益趨勢、預測用戶某段時間的投資組合收益等，以提升嵌入的有效性。但需注意任務權重平衡，確保主任務表現。  
 - 透過**PyTorch**的模組化設計，我們可以方便地組裝上述結構。例如，PyTorch官方的 **TorchRec** 庫為雙塔架構提供了現成的模組和大規模佈局支持<sup>20</sup>。我們可以利用TorchRec的embedding bag、sharding等功能，高效地處理用戶和資產塔中的大規模類別特徵（如股票ID、用戶ID）和稀疏特徵<sup>21</sup>。這對未來擴展至千萬級用戶與商品的工業場景尤為有利。

## 模型訓練與評估

### 訓練資料與標註策略

採用監督式方式對推薦模型進行訓練。利用歷史記錄構造用戶對資產的**偏好標籤**：  
 - **正樣本**：若用戶曾與某資產有過正向互動，則視為一個正反饋樣本。例如用戶實際買入某股票、收藏某基金、點擊查看了資產詳情，都可認定該用戶對該資產有興趣（正樣本=1）。  
 - **負樣本**：對於用戶在觀察期間未互動過的資產，可視為潛在的負樣本（=0）。由於用戶不可能與所有資產互動，因此需要採用**負採樣**策略從未交互項中抽取一部分作為訓練的負例。可結合隨機抽樣與策略抽樣（如接受歡迎度抽取或難負樣本挖掘）來生成負樣本集<sup>22</sup>。在一個訓練mini-batch中，可以對每個正樣本隨機配對若干負樣本來形成訓練三元組(user, pos\_item, neg\_item)。  
 - **樣本權重（可選）**：若有些行為比其他更能表示偏好（例如「購買」可能比「點擊」表示更強的興趣），可對不同類型的互動賦予不同權重，或將問題視為有等級的正反饋進行優化。但本指引聚焦於二元偏好即可。

### 損失函數與優化

模型同時學習用戶塔和資產塔的參數，使正樣本的用戶嵌入與資產嵌入相似度提高、負樣本降低。我們可選擇以下兩種主要訓練目標：  
 - **Pointwise二元交叉熵損失**：將推薦視為二分類問題，預測 $\hat{y}_{u,i} = \sigma(u \cdot v)$ 接近1、負樣本接近0計算交叉熵損失。這種方法直觀，且可利用現有框架方便地計算。但需注意正負樣本比例不均衡，通常會採用負採樣平衡。  
 - **Pairwise排名損失（BPR loss）**：直接優化排序，使正樣本得分高於負樣本。典型做法是對每個(user, pos, neg)三元組，讓 $\hat{y}_{u, pos} = u \cdot v$ ， $\hat{y}_{u, neg} = u \cdot v'$ ，優化 $\sigma(\hat{y}_{u, pos} - \hat{y}_{u, neg})$ 的對數似然或其近似（如hinge loss）。**Bayesian Personalized Ranking (BPR) 損失**<sup>23</sup> 透過最大化 $\hat{y}_{u, pos}$ 與 $\hat{y}_{u, neg}$ 的差距來實現個性化排序，是隱式反饋推薦中常用的方法<sup>24</sup>。本系統傾向使用BPR損失，因其直接對排序優化，比點wise方法更能提升Top-K推薦質量，同時利用**in-batch**負採樣等技巧可高效訓練<sup>25</sup><sup>22</sup>。公式表達為： $\mathcal{L}_{BPR} = -\sum_i \ln \sigma(u \cdot v_i - u \cdot v_j)$ ，其中 $v_i$ 為正樣本， $v_j$ 為負樣本。  
 - **正則化**：為防止過度擬合，在損失中加入權重正則項（如 $L_2$ 正則）對嵌入向量和MLP權重進行約束。另外，可對過熱門的資產或用戶embedding加強正則，以免受少數大眾熱門項目偏差影響。

優化過程採用**mini-batch梯度下降**（如SGD或Adam）。建議使用**Adam優化器**以自適應調節學習率，加速收斂。同時可使用**學習率衰減策略**（如學習率在若干個epoch後縮減）提升最終效果。每次迭代隨機抽取一批用戶及其正負樣本計算梯度，更新用戶塔和資產塔的參數。訓練中評估loss在驗證集上的走勢並使用**Early Stopping**：若驗證loss多輪不再下降則提前停止訓練，防止過擬合。

## 評估指標

訓練完成後，需要以多種指標評估模型在推薦任務和投資績效上的表現：

**推薦準確性指標：**- **Precision@K（精確率）**：對每位用戶推薦Top-K資產，其中屬於用戶實際喜愛/購買的比例。Precision@K越高表示推薦結果相關性越強。 - **Recall@K（召回率）**：在用戶所有偏好資產中，Top-K命中多少比例。由於推薦系統更關注精確率和NDCG，這裡Recall僅作參考。 - **Hit Rate@K（命中率）**：與Recall類似，計算每個用戶的Top-K是否至少命中一個真實正樣本，然後對用戶取平均。這反映每個用戶得到有效推薦的概率。 - **NDCG@K（歸一化折損累積增益）**：考慮命中位置的評價指標，對Top-K列表中不同位置的命中給予不同權重（越前面權重越高）。NDCG@K是衡量排序質量的重要指標，特別關注前列推薦的有效性。以上Precision、Hit Rate、NDCG等可在驗證/測試集上離線計算<sup>26</sup>。例如，我們在測試集中計算模型的Hit Rate@5、NDCG@5等，以比較不同模型或不同超參數的效果<sup>27</sup>。在相關研究中，動態模型通過考慮時間因素顯著提升了NDCG等排序表現，說明引入序列信息能夠避免僅推薦用戶已持有的項目、提供更多元的推薦<sup>27</sup>。

**投資績效指標：**- **年化報酬率**：計算推薦資產在後續一段時間（例如未來一年）的模擬投資組合收益，換算成年化百分比。由於我們的推薦系統旨在提供實際投資建議，因此關注推薦資產的未來表現。可採用**回測**(Backtesting)方式：對於測試集中每個用戶，在特定測試時點將Top-K推薦資產視為新投資組合，觀察之後例如30天或更長周期內該組合的回報<sup>28 29</sup>。將這段期間的累積收益折算成年化。然後與用戶原先持倉或市場基準比較，評估推薦是否帶來超額收益<sup>29</sup>。 - **夏普比率**：衡量風險調整後報酬的指標，定義為(投資組合平均報酬 - 無風險利率) / 投資組合收益波動率。例如對於推薦的Top-K資產等權組合，計算其日收益標準差和平均收益，求夏普值。夏普比率高表示單位風險下獲得的超額報酬高。評估時比較用戶原投資組合和推薦組合在測試期的夏普比率變化<sup>29</sup>。我們期望推薦能提高用戶整體組合的夏普值，即在控制風險的同時提升收益<sup>30</sup>。實驗可記錄每個用戶在採納推薦前後組合夏普比率的差異，統計平均改善幅度<sup>30</sup>。 - **最大回撤（可選）**：推薦組合在測試期間的最大資產淨值跌幅百分比，這從負面角度評估風險。若推薦導致組合最大回撤明顯增大，說明風險提高了，需要調整模型讓其對風險偏好的約束更嚴格。

在科研實驗中，一般會同時報告**推薦效果**（如Precision/NDCG）和**投資績效**（如平均收益、夏普比率）的指標<sup>26 31</sup>。例如某研究顯示，一些純收益導向的模型推薦表現（NDCG）很低，而純偏好導向模型收益表現不佳，需要在兩者間取得平衡<sup>32</sup>。因此，我們的系統應致力於**提升NDCG等推薦指標的同時，確保推薦資產具有良好的收益潛力和風險控制**<sup>32</sup>。透過上述多維度評估，可以全方位衡量系統實用價值並指導迭代優化。

## 推薦結果生成與解釋

### 推薦清單與內容呈現

模型在預測階段，對於每個請求的用戶會產生一份排序後的資產清單（通常取Top-K，例如K=10）。**推薦清單**中的每一項包括： - **資產名稱與基本資訊**：如股票名稱及代碼、基金名稱及類型等，方便用戶識別。 - **關鍵指標摘要**：對於股票，展示如當前股價、近期漲跌幅、所屬產業；對於基金，展示如最新淨值、近一年回報率、風險等級。同時可標註資產是否屬於用戶熟悉或關注的範疇（例如用戶曾投資過同產業的資產）。 - **推薦理由**：簡要的自然語言描述，說明該資產入選推薦的主要原因。理由應該與用戶提供的偏好/目標和資產特徵相呼應，以提升說服力<sup>33</sup>。例如： - 「此基金近一年報酬率在同類型基金中排名前10%，且風險評級為中低，符合您的穩健型風險偏好。」 - 「該股票所在的科技產業正處成長期，公司最近財報EPS連續三季上升，呈現良好增長勢頭。」 - 「您過去偏好金融業股票，這支銀行股市盈率低於同行平均，有價值低估的跡象，適合價值投資目標。」 理由生成可以由**模板+數據填充**實現，模板依據不同觸發點（如「績效優異」、「符合風險偏好」、「題材契合」等）填入相應資訊。更高階的作法可訓練一個**輔助模型**根據用戶和資產特徵生成個性化解釋。引用圖譜中的知識也有助於生成**知識驅動的解釋**（例如指出基金持有的明星股票）。提供清晰的推薦理由能增進用戶對系統決策的理解和信任<sup>34</sup>。 - **預期績效指標**：對推薦資產給出幾項關鍵的預期或歷史績效指標，幫助用戶直觀比較。例如「近3年年化報酬：+15%；夏普比率：1.2；最大回撤：-8%」。這些指標可以是資產歷史表現（作為未來的參考），也可以是假設用戶投資該資產後的一些模擬結果（如基於風險收益評估的預測區

間）。重點在於用數據支撐推薦的可靠性，例如當推薦一組基金時，列表可以附帶每支基金的三年平均收益和波動，以便用戶權衡取捨。

## 解釋性與可視化輔助

為進一步提高推薦結果的透明度，可在文字說明之外加入簡單的圖表視覺化： - **風險-收益散佈圖**：將推薦的多個資產以點狀圖方式呈現其歷史或預期年化收益率（縱軸）對比波動率或風險等級（橫軸）。這能讓用戶一眼看到推薦清單中哪些屬於高風險高回報，哪些穩健低風險，以確認是否符合其預期。理想情況下，推薦的資產應該大多數落在**高回報且風險符合用戶偏好的象限**中。用戶風險承受度不同，這個圖的重點區域也應不同：如積極型用戶可接受偏右上區域（高風險高收益），保守型用戶則偏左下區域（低風險適中收益）。 - **偏好匹配熱力圖**：若我們將用戶的幾個主要偏好維度（例如產業偏好匹配度、風險等級匹配度、歷史表現匹配度）對每個推薦資產打分，可以繪製一個熱力矩陣，橫軸為推薦資產，縱軸為偏好維度，顏色深淺表示匹配程度。用戶可從中直觀了解每個推薦項在各方面與自己需求的吻合程度。例如某支推薦股票在「產業符合度」上是深色（用戶喜愛的產業），但在「風險等級」上顏色較淺（風險略高於用戶設定），那用戶就能有針對性地考量取捨。 - **歷史走勢對比圖**：針對推薦的資產，可以顯示其近期價格或淨值走勢折線圖（比如近半年走勢），並疊加相關的指標（如同類平均走勢、市場指數走勢作對比）。這有助於用戶理解推薦資產是否最近處於上升趨勢、波動如何，以及相對表現如何。

透過上述可視化，結合文字理由，可以形成一個**多模態的推薦解釋界面**，增強用戶對系統決策的理解和對結果的接受度<sup>33</sup>。研究指出，提供解釋能提升用戶滿意度和信任，尤其在金融領域有助於合規與用戶教育<sup>34</sup>。

## 用戶互動與即時更新

系統部署在應用中後，應允許用戶透過交互界面調整個人偏好設定並立即獲取更新的推薦： - 當用戶修改了**投資目標**（例如從「追求成長」改為「保值避險」）或**風險偏好**等資料時，前端將新資料傳至後端推薦服務。服務會實時將新的用戶特徵餵入用戶塔重新計算用戶嵌入向量\$u'\$. 由於物件塔的資產向量\$v\$可以預先離線計算好<sup>18</sup>，在線只需對比\$u'\$與已有資產庫的向量集合，快速求取新的Top-K推薦<sup>19</sup>。利用高效的ANN索引（如FAISS）可在毫秒內完成這一近似最近鄰搜尋<sup>35</sup>。 - 若用戶提供了新的興趣標的（比如加入關注某股票），系統也能及時將此作為正反饋，調整推薦結果。實務上，可對用戶剛關注的股票給予一定提升，或將其所屬行業/類型作為偏好強化，在新一輪推薦時反映出來。 - 對於**新用戶**（無歷史），系統可引導其填寫一份風險偏好問卷或選擇幾個感興趣的行業/基金類型，以獲取初始偏好設定。這些信息將直接用於上述冷啟動推薦計算。隨著後續逐步蒐集到該用戶的互動行為，可定期或實時地微調用戶嵌入，使推薦越來越精準。 - 系統也應當定期更新資產資訊和模型參數。例如每日收盤後更新股票財務和價格特徵、每月更新基金排名，並利用最新數據對模型做小規模fine-tune或嵌入刷新，以保持推薦結果的時效性。因為金融市場變化快，模型需要不斷學習新的趨勢；若發現原模型對近期資料誤差增大，可啟動重新訓練流程或在線學習機制（如使用用戶新點擊作為增量學習信號）來調適。

## 開發與部署建議

本系統採用PyTorch實現，開發者可根據上述設計逐步構建並調試模型原型。在實作過程中需注意： - **數據預處理**：對不同來源的數據進行清洗對齊。確保用戶與資產特徵都有合理的數值範圍與填補缺失。對時間序列數據，統一頻率（如交易日）並進行必要的對數變換或差分以去除趨勢。 - **模型微調**：初始訓練完成後，可能需要通過調整嵌入維度、隱層大小、正則強度等超參數來優化模型效果。可以使用驗證集上的NDCG@K或Hit Rate@K作為指標進行調參。同時觀察投資績效指標，以在個人化與收益之間取得平衡<sup>32</sup>。若發現模型過於偏好熱門高收益資產而忽略用戶偏好，可以在loss中加入對用戶偏好匹配的正則或多任務學習來改善。 - **性能與擴展**：在有大量用戶和資產時，訓練和推理的性能要關注。利用PyTorch的**mini-batch並行**、GPU加速來加快計算。對於嵌入矩陣極大的情況，可採用**分布式訓練**或TorchRec的分片訓練方法<sup>21</sup>。線上部署時，重點在於使用ANN檢索加速Top-K查找<sup>35</sup>以及對模型做推理圖優化（如Tracing或Script提高PyTorch推理速度）。 - **模型監控**：部署後應監控推薦點擊率、轉化率等實際業務指標，以及組合收益的實現情況。如果發現與線下評估偏差較大，要分析原因（市場行情變化、用戶行為改變等）並迭代模型。

按照本指南實作的股票與基金推薦系統，將能夠根據用戶獨特的偏好和風險屬性，從台股、美股市場的海量投資標的中智能篩選出匹配的股票或基金產品，並透過雙塔模型高效產生推薦結果。在滿足準確性的同時，系統注重風險控管與多樣化，為每個推薦提供透明的理由和數據支持，最終輸出**Top-K清單、推薦原因以及績效評估**等完整資訊，輔助投資者做出更明智的決策<sup>1</sup><sup>34</sup>。透過不斷的數據更新與模型微調，該系統能逐漸學習市場的新動態和用戶的新偏好，成為用戶在投資理財道路上的可靠助手。以上即為開發者在PyTorch環境下進行模型fine-tuning或構建原型時的詳細實作藍圖。

---

<sup>1</sup> <sup>2</sup> <sup>23</sup> <sup>24</sup> <sup>26</sup> <sup>27</sup> <sup>28</sup> <sup>29</sup> <sup>30</sup> <sup>31</sup> <sup>32</sup> Stock Recommendations for Individual Investors: A Temporal Graph Network Approach with Mean-Variance Efficient Sampling

<https://arxiv.org/html/2404.07223v2>

<sup>3</sup> <sup>4</sup> <sup>8</sup> Incorporating Domain-Specific Traits into Personality-Aware Recommendations for Financial Applications | New Generation Computing

<https://link.springer.com/article/10.1007/s00354-024-00241-w>

<sup>5</sup> <sup>6</sup> <sup>33</sup> <sup>34</sup> Explainable mutual fund recommendation system developed based on knowledge graph embeddings | Applied Intelligence

<https://link.springer.com/article/10.1007/s10489-021-03136-1>

<sup>7</sup> <sup>12</sup> <sup>13</sup> <sup>14</sup> <sup>15</sup> <sup>17</sup> <sup>18</sup> <sup>19</sup> <sup>22</sup> <sup>25</sup> <sup>35</sup> The Two-Tower Model for Recommendation Systems: A Deep Dive | Shaped Blog

<https://www.shaped.ai/blog/the-two-tower-model-for-recommendation-systems-a-deep-dive>

<sup>9</sup> <sup>10</sup> <sup>11</sup> <sup>16</sup> DR-GAT: Dynamic Routing Graph Attention Network for Stock Recommendation

[https://www.researchgate.net/publication/375424854\\_DR-GAT\\_Dynamic\\_Routing\\_Graph\\_Attention\\_Network\\_for\\_Stock\\_Recommendation](https://www.researchgate.net/publication/375424854_DR-GAT_Dynamic_Routing_Graph_Attention_Network_for_Stock_Recommendation)

<sup>20</sup> <sup>21</sup> Introducing TorchRec, a library for modern production recommendation systems – PyTorch

<https://pytorch.org/blog/introducing-torchrec/>