Introduction

This report presents our approach to developing a personalized recommendation system for matching individual clients with the best financial advisors. Our model is designed to provide customized recommendations by taking into account the unique needs, preferences, and demographic profiles of each client. More specifically, our objectives are to:

- **Identify the most suitable financial advisor for each client:**
  We leverage historical data and advanced techniques to predict which advisor is most likely to meet the client's financial goals and needs.
- **Understand the key factors influencing client–agent matching:**
  Our analysis focuses on discovering the role of client demographics, advisor performance, and product expertise in driving successful client-advisor interactions.
- **Optimize client–agent pairing using relevant attributes:**
  By integrating deep metric learning with traditional feature engineering (e.g., product conversion rates and demographic similarities), we aim to create a model that maximizes the likelihood of a successful and productive advisor-client relationship.

Dataset Transformation

We conducted exploratory data analysis (EDA) on the three datasets and applied necessary transformations, including dropping, modifying, and adding columns to better suit our analysis.

Agent Dataset

The agent dataset contained a mix of demographic attributes (e.g., **agent_gender**, **agent_marital**) and experience-related attributes (e.g., **agent_tenure**, **cnt_converted**, **pct_sx1_male**). We applied several preprocessing steps to clean and standardize the data.

Preprocessing Steps:

1. Dropped Irrelevant Columns:
    - Removed **pct_SX0_unknown** and cluster as they were not relevant to our analysis.
2. Feature Transformations**:**
    - Gender Encoding**:** Converted **agent_gender** into a binary column, **agent_female**, where 1 represents female agents and 0 represents others.
    - **One-Hot Encoding:** Applied MultiLabelBinarizer to transform **agent_product_expertise** into separate one-hot encoded columns, which were concatenated with the original dataset.
3. Data Validation and Standardization:
    - Ensured that percentage distribution columns (e.g., **pct_AG01_1to20**, **pct_AG02_20to24**, ..., **pct_AG10_60up**) sum to 1 within each row.
    - Standardized **agent_marital** by mapping abbreviations (M, S, U, D, W) to full category names (Married, Single, Unknown, Divorced, Widowed).

Client Dataset

We validated the dataset to ensure consistency and prevent data type mismatches in later analyses.

Preprocessing Steps:

1. Data Type Corrections:
    - Converted numerical columns (**family_size, economic_status, household_size**) to integer (int64) format.
2. Address and Postal Code Cleaning:
    - Verified that valid Singapore postal codes range between 00XXXX and 82XXXX.
    - Identified 323 invalid cases, including codes such as 99XXXX, 5-digit codes, alphanumeric values, or missing entries, and removed them due to their small proportion.
    - Mapped valid postal codes to 28 districts, following official government-provided groupings.
3. Handling Missing Data:
    - Removed 30 rows with missing values in **DOB**, **Race**, and **economic_status**.
4. Feature Encoding and Transformations:
    - Ordinal Encoding:
        - **household_size_grp**: HH1_1t40 → 1, HH2_40to80 → 2, and so on.
        - **family_size_grp**: FS1_1t20 → 1, FS2_20to40 → 2, and so on.
    - Marital Status Standardization:
        - Mapped **client_marital** to full names (Married, Single, Unknown, Divorced, Widowed), ensuring consistency with the Agent Dataset.
5. Dropped Redundant Columns:
    - Removed **household_size** and **family_size**, as they were already summarized in **household_size_grp** and **family_size_grp**.

Policy Dataset

Preprocessing Steps:

1. Date Conversion and New Feature Creation:
    - Converted **occdate** to datetime format.
    - Created a new column, **policy_age**, by computing the difference between 2025 and the year extracted from **occdate**.
2. Dropped Unnecessary Columns:
    - Removed **occdate, flg_main, flg_rider, flg_inforce, flg_cancel, flg_converted**, and **product_grp**, as they were not required for our analysis.

Methodology

Feature Engineering

Feature engineering is an important step in developing machine-learning models.
By carefully designing and selecting features, we can discover hidden patterns, improve model accuracy, and ensure better generalization to unseen data.

Custom Features/Variables to improve our model

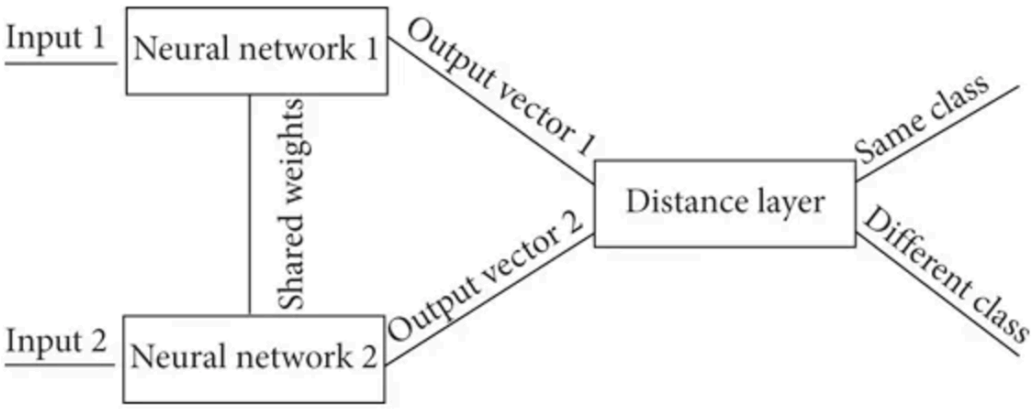
match_count	How often the agent was chosen for similar clients?
gender_score	The similarity in gender between the agent's typical clients and the new client.
age_score	The similarity in age between the agent's typical clients and the new client
agent_product_rating	How well the agent has sold relevant products.
back_ground_similarity	The cosine similarity score for demographics.

match_count

This feature represents the number of successful matches associated with each agent. It is calculated by aggregating the number of matches per agent and filling missing values with zero. It is also used as a dependent variable in our regression model.

gender_score

It is computed as the proportion of matches with male **(pct_SX1_male**) and female **(pct_SX2_female**) clients. This feature helps capture potential gender-based preferences in match success.



To train the Siamese network, positive and negative pairs of clients were created. Positive pairs consisted of clients served by the same agent, while negative pairs consisted of clients who were never served by the same agent.

A Siamese neural network was trained to learn a 16-dimensional embedding for each client. The network was trained using contrastive loss, which ensures that similar clients are close together in the embedding space, while dissimilar clients are far apart. These embeddings allow us to find the most similar clients for a new client in the embedding space

The products most frequently purchased by these similar clients were identified, and a suitability score was calculated for each product. For each agent, a score was calculated based on their historical conversion rates and expertise with the top products. This score was normalized to ensure consistency across agents.

back_ground_similarity

The similarity between the new client's demographics (age, gender, marital status) and the agent's typical client profile was calculated using cosine similarity. This feature captures how well the agent's experience aligns with the new client's background.

Prediction using XGBoost

Our XGBoost model is used to predict the best financial advisor (agent) for a client by combining historical interactions, demographic similarities, and product expertise. The system uses two different approaches:

1. **Classification Model: Predicting a Successful Match**

The classification model is designed to determine whether an agent is likely to match successfully with a new client. The system considers an agent to be a successful match if they have previously worked with similar clients.

To train this model, we first define a target variable, which indicates whether an agent has historically been assigned to clients similar to the new client. If an agent has successfully worked with a similar client before, they receive a positive label (1), and if not, they receive a negative label (0).

The dataset is then split into training and testing sets, ensuring that 80% of the data is used for learning, while 20% is reserved for testing the model's accuracy.

The XGBoost classifier is trained on key features such as:

- The number of previous matches an agent has had with similar clients.
- The agent’s success in selling relevant financial products.
- The demographic similarity between the agent’s past clients and the new client.

Once trained, the model predicts the probability that an agent will successfully match with a new client. Agents are then ranked based on these probabilities, with the highest-scoring agents being recommended first.

To evaluate the model’s accuracy, we use the Area Under the Curve (AUC) metric, which measures how well the model distinguishes between successful and unsuccessful matches. We also analyze feature importance, which allows us to understand which factors are most influential in determining agent success.

2. Regression Model: Predicting Expected Match Count

While the classification model predicts whether an agent will match successfully, the regression model estimates how many successful matches an agent is expected to have. Instead of using a binary success/failure approach, this model predicts a continuous number of matches based on an agent's historical performance.

The target variable for this model is the match count, representing the number of times an agent has successfully worked with similar clients in the past.

As with the classification model, the dataset is split into training and validation sets to ensure the model generalizes well to new data.

The XGBoost regression model is trained on the same set of features, including:

- Agent experience with similar clients.
- Past success rates in selling relevant products.
- Demographic alignment with potential clients.

Once trained, the model predicts how many successful matches each agent is likely to have, allowing us to rank agents based on expected performance rather than just probability.

The model's accuracy is measured using Root Mean Squared Error (RMSE), which assesses how close the predicted number of matches is to actual values. Feature importance analysis is also performed to determine which factors contribute most to an agent's ranking.

3. Final Agent Ranking & Recommendation

Once both models have been trained, we generate a final ranking of agents using their predicted scores. The ranking is based on either:

- The probability of a successful match (from the classification model).
- The expected number of successful matches (from the regression model).

Agents with the highest scores are prioritized in recommendations, ensuring that the system suggests the most suitable financial advisors for each client.

Results

Our analysis reveals that several features are highly significant predictors in our recommendation system. In particular, our feature importance analysis indicates that:

- **match_count:** The frequency with which an agent has served clients similar to the new client.
- **gender_score:** The degree of alignment between the agent's historical client gender distribution and the new client's gender.
- **age_score:** The similarity in age distribution between an agent's historical clients and the new client.
- **Agent–Product Rating:** A composite metric that combines the agent's conversion percentages and product expertise for products that are most relevant to the new client.
- **Background Similarity:** The cosine similarity between the new client's demographic profile (age, gender, marital status) and that of the agent's typical clients.

Our models achieved strong performance on our validation set:

- **XGBoost Classifier:**
  The classifier achieved a validation AUC near 1.0, indicating excellent discrimination between agents who have a history of successful matches and those who do not.
- **XGBoost Regressor:**
  The regression model, which predicts the actual match count, achieved a low RMSE of 0.12. This low error indicates that our model can accurately predict the expected number of successful matches for each agent.

These results demonstrate that our hybrid approach—combining metric learning, product expertise, and demographic similarity—effectively identifies and ranks agents likely to perform well with a new client.

Addressing Ethical Issues & Bias

Our model uses demographic features (such as age, gender, and marital status) to improve the quality of agent-client matching. However, we have taken several steps to ensure that these features are used ethically and do not lead to discriminatory outcomes:

- **Sensitive Feature Handling:**
  Although demographic features are used to enhance matching quality, we intentionally removed features such as race to prevent any potential bias in the recommendations.
- **Fairness in Recommendations:**
  Agents who have not been historically matched (i.e., with a match_count of 0) are not automatically assigned a zero probability of being recommended. Instead, our scoring system ensures that every agent has a fair chance to be matched based on their overall performance, product expertise, and background similarity.
- **Transparency:**
  Every data transformation and modelling decision is thoroughly documented, ensuring transparency in how the recommendations are generated. This documentation is part of our submission and aims to foster trust in the system.

Through these measures, we aim to ensure that our recommendation system is both effective and fair, providing personalized advisor assignments without introducing unintended biases.

Insights and Conclusion

Our analysis highlights **agent–product rating** and **background similarity** as key factors in ranking agents. Agents who excel in selling relevant products and serve clients with similar demographics tend to perform better.

The model is **scalable and efficient**, leveraging precomputed client embeddings and cosine similarity for real-time recommendations. By combining **metric learning, product expertise, and demographic similarity**, our system personalizes advisor assignments to match client needs, improving satisfaction and business outcomes.