NUS DATATHON 2025

TEAM NUMBER: 34
TEAM NAME: Team BUFF
TEAM MEMBERS: Zhang ZiZhong, Yao William, Wu Chia-tung, Chen Ping

Matching clients with agents with Python

In this project, we aim to optimize the assignment of financial advisors to customers, achieving the following two key objectives for Singlife:

1. Maximise Revenue
2. Ensure fairness and ethical business practices

Instructions for Setting Up the Environment

To run this project, you need a **Python environment** with the following libraries installed:

- `pandas` (for data manipulation)
- `numpy` (for numerical computations)
- `scipy` (for similarity calculations)
- `scikit-learn` (for machine learning utilities)

If using **Google Colab**, the required libraries are pre-installed. Otherwise, install them using:

pip install pandas numpy scipy scikit-learn

How to Run the Notebook and Reproduce Results

1. Open the **Colab notebook**: Project Link.
2. Upload the required datasets:

- `agent_info_df = pd.read_parquet(write your file path for nus_agent_info_df here)`
- `client_info_df = pd.read_parquet(write your file path for nus_client_info_df here)`
- `policy_info_df = pd.read_parquet(write your file path for nus_policy_info_df here)`
- `sample_final_df = pd.read_parquet(write your file path for sample_final_modelling_df here)`

3. Run the cells **sequentially** to:
   - Load and preprocess data.
   - Train the feature similarity-based recommendation model.
   - Run the second Comprehensive Agent Scoring Model
   - Generate agent recommendations for clients.
   - Evaluate model performance by comparing predicted matches with historical data.

Specific Instructions Required for Executing the Model

Execution Order: The FWES model must be run before the Comprehensive Agent Scoring Model.

Sample final output:

```
✅ Final Top 5 Recommendations (with Cancellation Rate Adjusted):
      secuityno   agntnum  combined_score  pct_cancel
0         CIN:0   AIN:711        1.253463       0.014
1         CIN:0    AIN:53        0.857138       0.133
2         CIN:0   AIN:521        0.686212       0.179
3         CIN:0  AIN:1029        0.614269       0.000
4         CIN:0   AIN:854        0.180256       0.058
...         ...       ...             ...         ...
99390  CIN:9999   AIN:711        0.429421       0.014
99391  CIN:9999  AIN:1029        0.347697       0.000
99392  CIN:9999   AIN:854        0.006182       0.058
99393  CIN:9999   AIN:669       -0.017308       0.065
99394  CIN:9999   AIN:623       -0.111780       0.089

[99395 rows x 4 columns]
```

Our Approach

EDA

Data Cleaning

Imputation Techniques

Feature Engineering

Feature-Weighted Euclidean Similarity (FWES)

Comprehensive Agent Scoring Model

Data Visualization

To gain initial insights into the dataset, we visualized key features using histograms and scatter plots to understand data distributions and relationships between variables. These visualizations helped us to achieve the following goals:

1. Detect data patterns
    - Understand the distribution of key variables like **economic status** among clients (figure 1)
    - Analyze the relationship between variables like **agent tenure** and **converted policies** (figure 2)

2. Identify Data Issues:
    - Spot missing values and outliers
    - Recognize skewed distributions

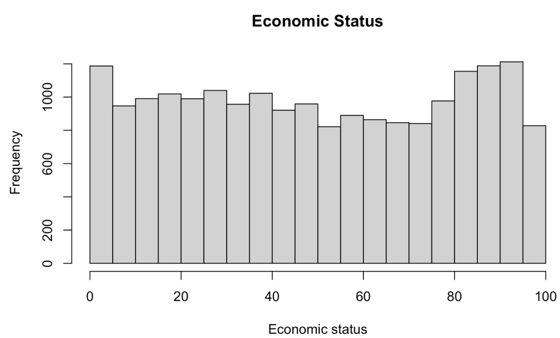

Figure 1: Economic Status

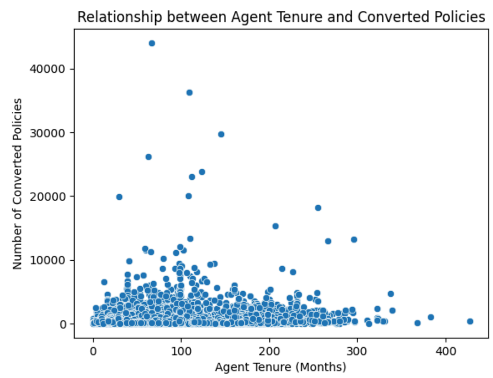

Figure 2: Agent Tenure and Converted Policies

Data Cleaning

- **Handling Missing Values**
    - Numerical columns with missing values are replaced with the **median** instead of the mean, as the median is less affected by outliers and maintains the integrity of data distribution.
    - Categorical columns (e.g. **race-desc_map** and **cltpcode**) are removed because the median imputation method cannot be applied.

- **Ethical considerations for data imputation**
    - One team member suggested replacing missing values for race with the most common value in the dataset. However, we decided against this approach because it would be unethical to assume a client's race as this could introduce bias into the model.

Data Preprocessing

- **Feature Engineering**
    - We created a new column "age" by subtracting each client's date of birth from a reference date (2025/02/02)
    - This ensures that age – a potential key factor in financial advisory – is thoroughly considered in our machine learning later.

- **Data Type Conversion**
    - Columns like **household_size, economic_status, and family_size** were stored as **objects** instead of **numeric** values.
    - We used the pd.to_numeric() to convert these variables to float, allowing for proper numerical processing.

- **One-hot Encoding for Categorical Data**
    - Some columns like **gender** were categorical variables. We applied one-hot encoding by converting them to binary values (0 or 1) to make them compatible with our model.

- **Preventing Overfitting from Duplicate Purchases**
    - We noticed that some clients have purchased the same policy multiple times on the same day, possibly for their family members, as shown below.
    - To prevent the problem of overfitting, we have aggregated duplicate transactions by summing the total policy purchases amount for each customer on a given day.

```
PID:520,AIN:131,CIN:18680,2019-12-17,80.0,prod_6,1,0,1,0,0,0,1,PG:0,AG07_45to49,TNR2_lt1yr
PID:521,AIN:131,CIN:18680,2019-12-17,37.0,prod_6,1,0,1,0,0,0,1,PG:0,AG07_45to49,TNR2_lt1yr
PID:519,AIN:131,CIN:18680,2019-12-17,80.0,prod_6,1,0,1,0,0,0,1,PG:0,AG07_45to49,TNR2_lt1yr
PID:522,AIN:131,CIN:18680,2019-11-05,37.0,prod_6,1,0,1,0,0,0,1,PG:0,AG07_45to49,TNR2_lt1yr
PID:517,AIN:131,CIN:18680,2019-12-17,37.0,prod_6,1,0,1,0,0,0,1,PG:0,AG07_45to49,TNR2_lt1yr
PID:518,AIN:131,CIN:18680,2019-11-05,80.0,prod_6,1,0,1,0,0,0,1,PG:0,AG07_45to49,TNR2_lt1yr
```

ML Model, Feature Selection, & Results

The **Feature-Weighted Euclidean Similarity (FWES) Algorithm** is a content-based recommendation system designed to match insurance clients with the most suitable financial advisors. FWES uses a weighted Euclidean distance metric to quantify the similarity between a client's attributes and the historical profiles of agents' serviced clients. By featuring importance scores and ranking agents by its similarity, this algorithm aims to improve the precision of recommendations and optimize client-agent pairings.

```
True Positives (TP): 519
False Positives (FP): 4780
False Negatives (FN): 4932
True Negatives (FN): 53643082
Precision: 0.10
Recall: 0.10
F1 Score: 0.10
Accuracy: 1.00
```

Next Steps

If we were given more time for this datathon, we would like to take following approaches:

- Include the Fairness-aware ML technique to ensure that model recommendations do not disproportionately exclude certain client groups
- Develop a feedback system where clients can rate advisors to refine future recommendations.