Price Setter and Follower Detection

Pricing, one of the most important aspects for business!

Problem Statement

redBus is an online platform where bus operators offer their services and sell seats. These bus operators vary from single service operators to ones which have scores of services.

Pricing is a complex decision for Bus Operators, who continuously adjust the prices seeing the demand and supply signals. But not all operators have the same level of visibility on these signals or the capability to price effectively. Operators who have better ability/confidence may price independently while some others may choose to follow the prices of such price setters.

What we need is a data based method to reveal these market dynamics.

Objective

Your challenge is to find out which services is/are pricing independently (the price leaders) and which services are the followers (and following whom?), given the history of prices set by the operators.

Data

There are two types of seats, but within one type also there can be multiple prices for different categories (front/back/upper/lower etc.) of seats. This categorization is decided by the operator based on the bus, hence there may be different numbers of prices for different services.

Data Fields

1. **Seat Fare Type 1** – Within Seat Type 1, the prices of all categories of available seats as defined by the operator.
2. **Seat Fare Type 2** - Within Seat Type 2, the prices of all categories of available seats as defined by the operator.
3. **Bus** – A particular bus service, for example, Hyderabad to Pune Go Tours 9:15 PM bus.
4. **Service Date** – The date of journey for which the prices are recorded.
5. **Recorded At** - The time when prices were recorded.

Evaluation

We have an idea of the market dynamics and we know the price leaders, who follow whom etc. The winning entry should come up with a data based approach to identify price leaders and followers.

For each bus service, the code should calculate who this particular service follows, and which service is the closest follower of this service. If the algorithm calculates the confidence score for such identification, the score should also be given. The format of the CSV file is given.

We will judge the entries based on how many leaders/followers/independents the code can correctly identify.

Submission

There are three things that you need to submit.

1. The python script named <teamID_teamName_code.py> which takes as input the path of the dataset, implements the algorithm and saves the output csv file.
2. Submit the output csv file with the name <teamID_teamName_output.csv>.
3. Submit a one-page PDF report named <teamID_teamName_report.pdf> which should contain the summary of your approach.

Note: Follow the naming conventions strictly, otherwise your submission will not be evaluated.