Fine_Tuners

Kharagpur Data Science Hackathon

Members

Aditya Ganguly
Tanvir Alam
Sayan Acharya
Anurup Naskar
Ritodeep Sikdar

Task Description and Dataset

In this task, we were given a csv file where various fields like bus_id, seat type 1, seat type 2, recording time, service date were mentioned. In those seat type 1 various prices of seats like: front, back, side etc were given. We were supposed to use that data to predict which bus_id was following exactly which bus_id with a confidence score.

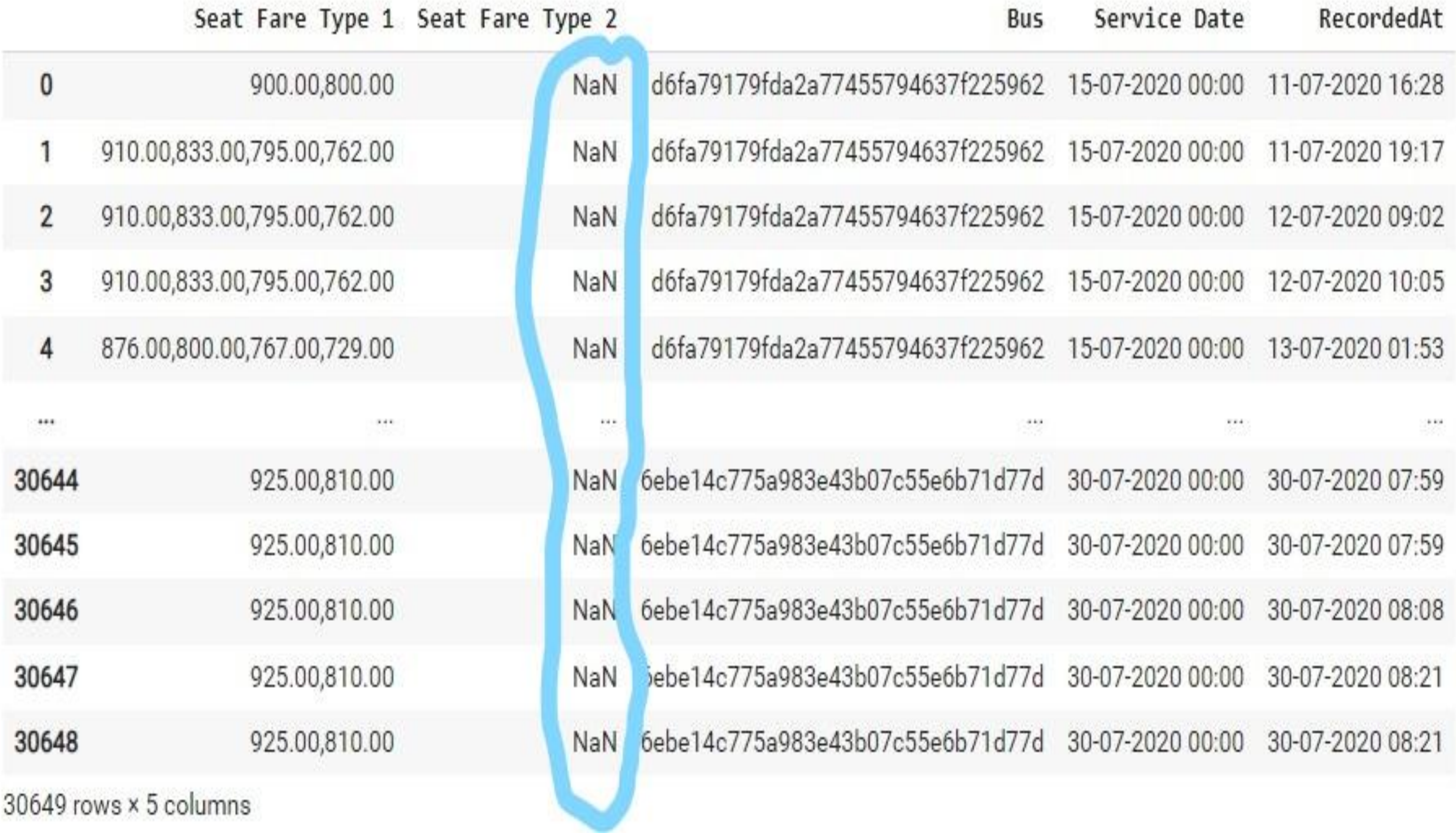
Intuition behind this problem

We cleaned the dataset and plotted the graphs for respective bus_id s . Then we found out that for a lot of buses, there was an uncanny similarity in the shape of the graphs. We realized that this idea of similarity can be used to find out the follower-leader pairs of these buses. Now we saw that the graphs are not always a hundred percent match and also the graphs do not start from the same point all the time. For that we came up with an idea (later discussed).

Brief Overview

We performed a cursory cleaning of the data.Then we used basic mathematical operations like polynomial interpolation and longest common subsequence. Using this we are able to.

1. Data Cleaning
2. Interpolation and graph drawing
3. LCS and graph matching
4. Prediction



	Seat Fare Type 1	Seat Fare Type 2	Bus	Service Date	RecordedAt
0	900.00,800.00	NaN	d6fa79179fda2a77455794637f225962	15-07-2020 00:00	11-07-2020 16:28
1	910.00,833.00,795.00,762.00	NaN	d6fa79179fda2a77455794637f225962	15-07-2020 00:00	11-07-2020 19:17
2	910.00,833.00,795.00,762.00	NaN	d6fa79179fda2a77455794637f225962	15-07-2020 00:00	12-07-2020 09:02
3	910.00,833.00,795.00,762.00	NaN	d6fa79179fda2a77455794637f225962	15-07-2020 00:00	12-07-2020 10:05
4	876.00,800.00,767.00,729.00	NaN	d6fa79179fda2a77455794637f225962	15-07-2020 00:00	13-07-2020 01:53
...	...	...	...	...	...
30644	925.00,810.00	NaN	6ebe14c775a983e43b07c55e6b71d77d	30-07-2020 00:00	30-07-2020 07:59
30645	925.00,810.00	NaN	6ebe14c775a983e43b07c55e6b71d77d	30-07-2020 00:00	30-07-2020 07:59
30646	925.00,810.00	NaN	6ebe14c775a983e43b07c55e6b71d77d	30-07-2020 00:00	30-07-2020 08:08
30647	925.00,810.00	NaN	6ebe14c775a983e43b07c55e6b71d77d	30-07-2020 00:00	30-07-2020 08:21
30648	925.00,810.00	NaN	6ebe14c775a983e43b07c55e6b71d77d	30-07-2020 00:00	30-07-2020 08:21

30649 rows × 5 columns

Data Processing

- ❑ We only consider the recorded dates in our analysis as we believe it is most indicative of the trends present in the data.
- ❑ First we clean the data, we remove all rows having nans and duplicate rows. When multiple prices are present for the same date, we consider the average.

- ❑ We then group the buses according to their serial number and further sort the data based on the dates.
- ❑ This reduced redundancy in data and then we used that data to plot the time-price graph. Then we used polynomial interpolation to get a polynomial best fitting for the graph.

Interpolation

- We took all the points and for the points of a single day, aggregated them together by taking average of them.
- Then we created (Day, Price) pairs for all these points
- Then we created the scatter plot for these points
- Now, we calculated the polynomial interpolation to fit the points in our curves.

We had previously tried with curve fitting (Regression) as well but the number of points per bus was not very much the same and also many buses had points that were quite far away from each other, so regression did not give a satisfactory value. So we resorted to polynomial interpolation which turned out to give better solution.

LCS and Graph Matching

Then we calculated the derivatives of that polynomial to help us understand exactly how the data is changing from date to date. This is crucial. Because, a follower and a leader may not start at the same date.

So, we detected similar patterns of changing of data with derivatives. Then we calculated the Longest Common Subsequence (LCS) to calculate the similar patterns. According to the match there, we put a score for every bus pair. If the initial recording time is greater then we considered them as leaders. Then leaders are selected according to who started the change. This method gave us very satisfactory result.

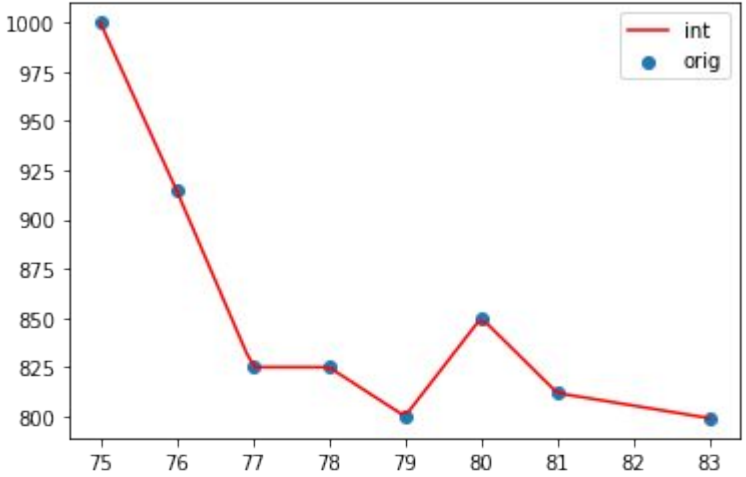
Scale

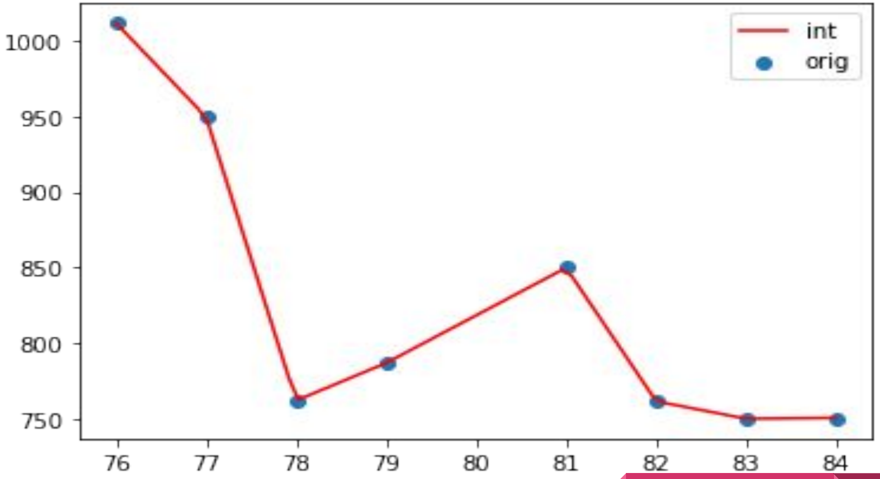
Y axis: The price of the seats in Rs

X axis: The days from august-july are converted into integers like, the first day of august is 1, 1st of june is 31+1=32 and so on.

Graphs and Analytics

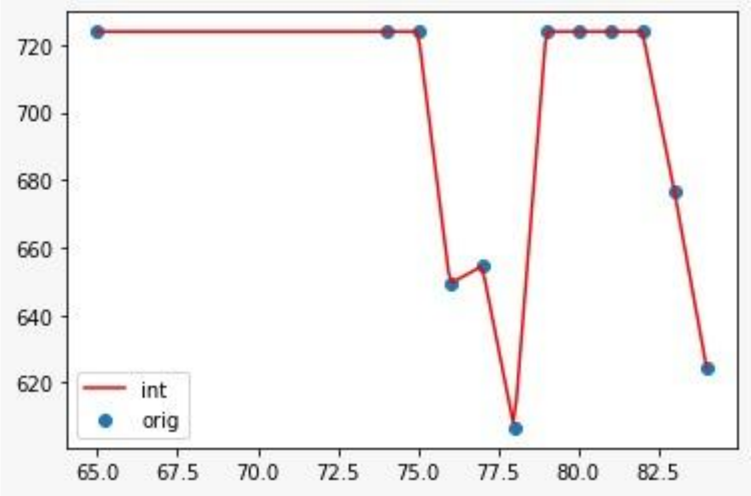
Matching Graphs : Example I

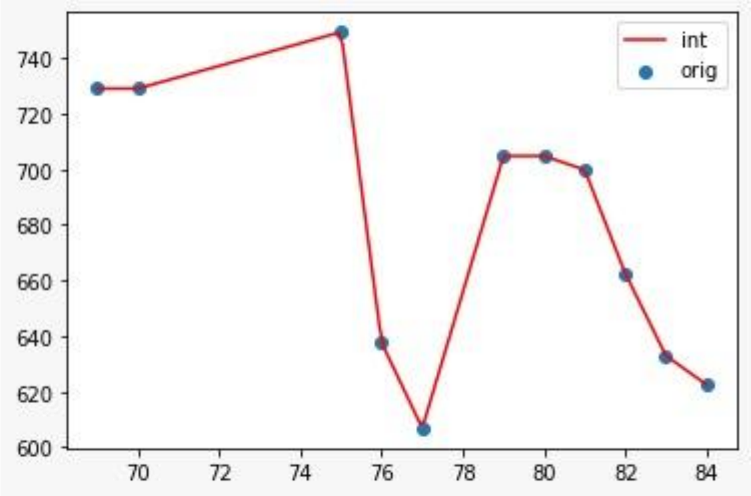




Here,we have shown two graphs that are practically similar.

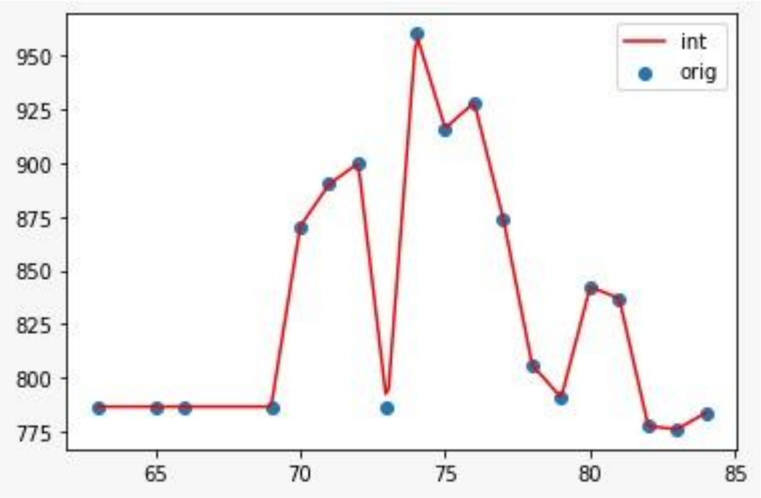
Example II : Matching Graphs

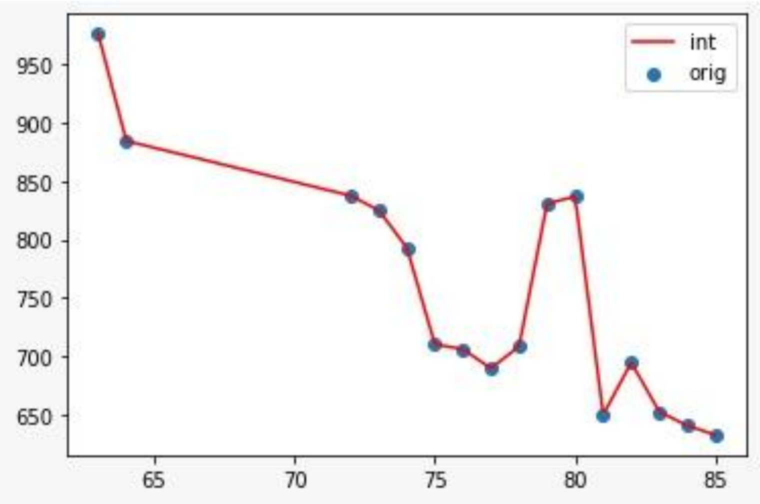




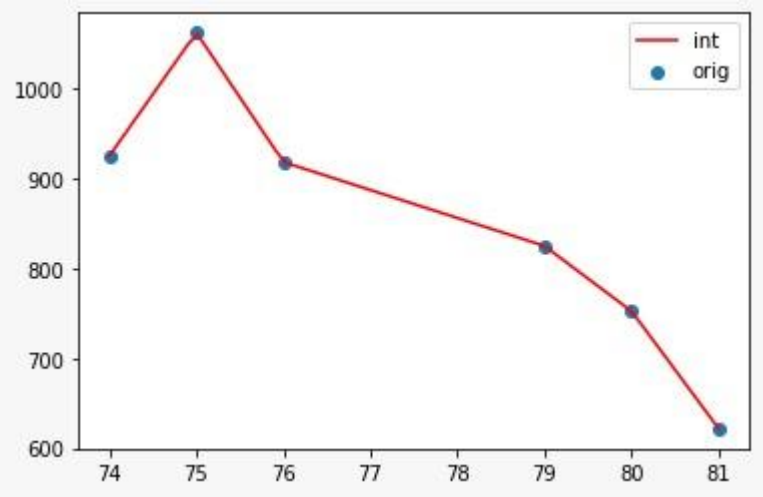
Smart Graph Matching

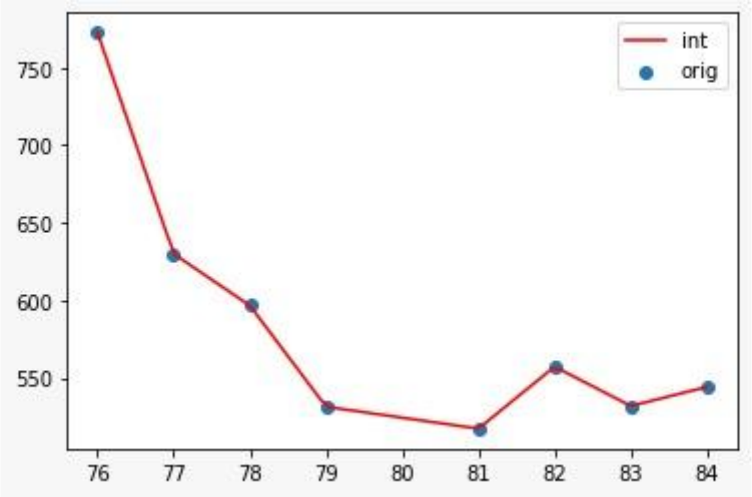
These two graphs look totally dissimilar at the first look. When we look closely, we can see that the part after the global maxima of the 1st graph is almost similar to the whole 2nd graph (maybe a bit stretched but more or less the same). This is happens as LCS was able to pick up similarities between subsections of the graphs. This can help when the two graphs are following each other for some portions.





Example Of Non-Matching Graphs





Prediction

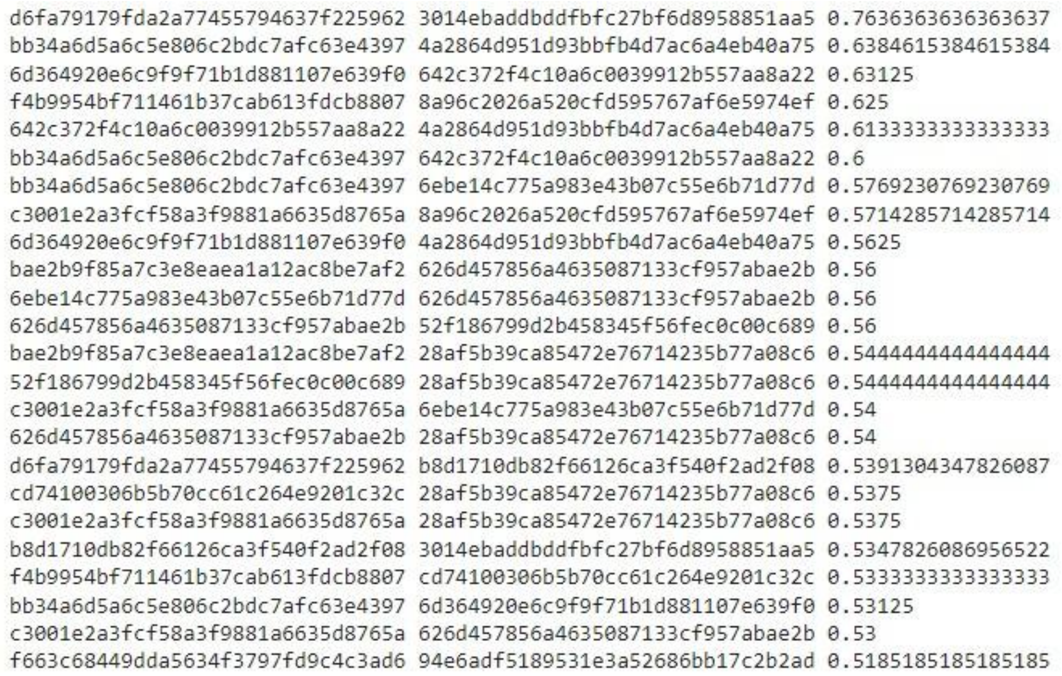
We used the LCS data for the prediction. Though there are some inaccuracies over the whole result but it was mostly satisfactory. We should consider the buses with a confidence score > 0.5 to be follower, leader pair.

Some pairs are seen to have a confidence score of near 1. This is some form of over-fitting.

With these mind, we can effectively set the valid range of confidence scores of 0.4 to 0.7

EXPERIMENT DETAILS AND OBSERVATIONS

This is a snapshot of the result of our findings.



```
d6fa79179fda2a77455794637f225962 3014ebaddbddfbfc27bf6d8958851aa5 0.7636363636363637
bb34a6d5a6c5e806c2bdc7afc63e4397 4a2864d951d93bbfb4d7ac6a4eb40a75 0.6384615384615384
6d364920e6c9f9f71b1d881107e639f0 642c372f4c10a6c0039912b557aa8a22 0.63125
f4b9954bf711461b37cab613fdcb8807 8a96c2026a520cfd595767af6e5974ef 0.625
642c372f4c10a6c0039912b557aa8a22 4a2864d951d93bbfb4d7ac6a4eb40a75 0.6133333333333333
bb34a6d5a6c5e806c2bdc7afc63e4397 642c372f4c10a6c0039912b557aa8a22 0.6
bb34a6d5a6c5e806c2bdc7afc63e4397 6ebe14c775a983e43b07c55e6b71d77d 0.5769230769230769
c3001e2a3fcf58a3f9881a6635d8765a 8a96c2026a520cfd595767af6e5974ef 0.5714285714285714
6d364920e6c9f9f71b1d881107e639f0 4a2864d951d93bbfb4d7ac6a4eb40a75 0.5625
bae2b9f85a7c3e8eaea1a12ac8be7af2 626d457856a4635087133cf957abae2b 0.56
6ebe14c775a983e43b07c55e6b71d77d 626d457856a4635087133cf957abae2b 0.56
626d457856a4635087133cf957abae2b 52f186799d2b458345f56fec0c00c689 0.56
bae2b9f85a7c3e8eaea1a12ac8be7af2 28af5b39ca85472e76714235b77a08c6 0.5444444444444444
52f186799d2b458345f56fec0c00c689 28af5b39ca85472e76714235b77a08c6 0.5444444444444444
c3001e2a3fcf58a3f9881a6635d8765a 6ebe14c775a983e43b07c55e6b71d77d 0.54
626d457856a4635087133cf957abae2b 28af5b39ca85472e76714235b77a08c6 0.54
d6fa79179fda2a77455794637f225962 b8d1710db82f66126ca3f540f2ad2f08 0.5391304347826087
cd74100306b5b70cc61c264e9201c32c 28af5b39ca85472e76714235b77a08c6 0.5375
c3001e2a3fcf58a3f9881a6635d8765a 28af5b39ca85472e76714235b77a08c6 0.5375
b8d1710db82f66126ca3f540f2ad2f08 3014ebaddbddfbfc27bf6d8958851aa5 0.5347826086956522
f4b9954bf711461b37cab613fdcb8807 cd74100306b5b70cc61c264e9201c32c 0.5333333333333333
bb34a6d5a6c5e806c2bdc7afc63e4397 6d364920e6c9f9f71b1d881107e639f0 0.53125
c3001e2a3fcf58a3f9881a6635d8765a 626d457856a4635087133cf957abae2b 0.53
f663c68449dda5634f3797fd9c4c3ad6 94e6adf5189531e3a52686bb17c2b2ad 0.5185185185185185
```

Improvements and Accuracy improvement

Along with considering the LCS between two different bus pairs, we could calculate an objective function encapsulating the follower-leader relationship between the bus pairs using the above LCS method. This is equivalent to fine tuning the lcs method.

We then formulate a quadratic objective function with decision variables X[i][j] denoting probability that bus i follows bus j.

We then formulate the quadratic objective function as Σ (score[i][j] - X[i][j])**2 . We then minimize this quadratic objective using any standard qp solver. For a high score, the decision variables will increase to minimize the final objective

This maybe an improvement to the method which we have considered above.This idea can be further extended for solving the overfitting problem.

Apart from that, instead of matching from the beginning, we can analyze the matched LCS to see exactly where the most apparent match started. That can be used to predict a better follower and leader pair.

Generalizations

1. In the real world we often find that the **stock market** is influenced by isolated individuals, we can use the above method to find who are the leaders and who are the followers in this context.
2. In the **animal kingdom**, many animals exhibit group behaviour such as birds and microbes, in these groups we can find the leader of the group using the above approach.
3. Nowadays, **software duplication** is rampant. With this approach we can determine which developer is borrowing ideas from whom.
4. With some tuning of parameters and selection of features,

   we can use this approach **find out cheaters**.

Thank You !