***A Project Report***

*on*

**PRICE SETTER FOLLOWER DETECTION**

*carried out as part of the* ***Mini Project DS3101*** *Submitted*

by

**Pranjal Paira**

**209309016**

**Avval Kaur**

**209309001**

*in partial fulfilment for the award of the degree* *of*

**Bachelor of Technology**

in

**Data Science**

Under the Guidance of

**Guide Name**



**School of Computing and Information Technology**

**Department of Information Technology**

**MANIPAL UNIVERSITY JAIPUR**

**JAIPUR-303007**

**RAJASTHAN, INDIA**

**November 2022**

**CERTIFICATE**

Date: 22/11/2022

This is to certify that the minor project titled **PRICE SETTER FOLLOWER DETECTION** is a record of the bonafide work done by **Pranjal Paira** **(209309016)** and **Avval Kaur (209309001)** submitted in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Data Scienceof Manipal University Jaipur, during the academic year 2022-23.

**Dr. Shweta Sharma**

*Project Guide, Department of Information Technology*

Associate Professor & Project Guide

Department of IT

*Manipal University Jaipur*

**ABSTRACT**

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

The dataset provided by RedBus has explicitly two types of seats, but within one type also there can be multiple prices for different categories (front/back/upper/lower etc.) of seats.The data provided was preprocessed and a time series graph was plotted. The time series graph suggests the impact of each category that should be taken under consideration. Finally, the Dynamic Time Wrapping algorithm was used to calculate DTW distance that can provide us with a comparison between the buses. The minimum DTW distance was taken and stored that corresponds to the closest comparison list. Using this we're able to find who follows whom. Later the distance was passed to the sigmoid function to produce a confidence score. Finally, the data was stored in a data frame.

**Table of Figure:-**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Figure no.** | **Page no.** |
| 1 | Figure (1) | 6 |
| 2 | Figure (2) | 8 |
| 3 | Figure (3) | 8 |
| 4 | Figure (4) | 8 |
| 5 | Figure (5) | 9 |
| 6 | Figure (6) | 11 |
| 7 | Figure (7) | 12 |
| 8 | Figure (8) | 13 |
| 9 | Figure (9) | 14 |
| 10 | Figure (10) | 14 |
| 11 | Figure (11) | 15 |
| 12 | Figure (12) | 15 |
| 13 | Figure (13) | 16 |
| 14 | Figure (14) | 16 |
| 15 | Figure (15) | 16 |

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.No.** |  | **Page** |
| i) | Certificate | i |
| ii) | Abstract | ii |
| 1. | Introduction | 2-4 |
| 2. | Background details | 5 |
| 3. | Design and Implementation | 6-11 |
| 4. | Implementation and Result | 12-17 |
| 5. | Conclusion | 18 |
| 6. | References | 19 |

1. **INTRODUCTION**

A time series is nothing but a sequence of various data points that occurred in a successive order for a given period of time.

Time series analysis helps organizations understand the underlying causes of trends or systemic patterns over time. Using data visualizations, business users can see seasonal trends and dig deeper into why these trends occur. With modern analytics platforms, these visualizations can go far beyond line graphs.

When organizations analyze data over consistent intervals, they can also use time series forecasting to predict the likelihood of future events. Time series forecasting is part of predictive analytics. It can show likely changes in the data, like seasonality or cyclic behavior, which provides a better understanding of data variables and helps forecast better.

Deep learning methods offer a lot of promise for time series forecasting, such as the automatic learning of temporal dependence and the automatic handling of temporal structures like trends and seasonality.

In time series analysis, with respect to this project where DTW is used, dynamic time warping (DTW) is an algorithm for measuring similarity between two temporal sequences, which may vary in speed.

**1.1Problem 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.

**1.2Motivation**

The basis of this project was to construct a data-based method or algorithm that can reveal the market dynamics, which usually occurs in the price-setting game for a bus. This method should broadly classify which bus operator changes price based on supply and demand independently and who is a follower of other price setters.

The key motivation behind was to can to analyze time series data in two key ways: to generate inferences on how one or more variables affect some variable of interest over time, or to forecast future trends. Unlike cross-sectional data, which is essentially one slice of a time series, the arrow of time allows an analyst to make more plausible causal claims.

**1.3 Objective**

* The project aims , AI to learn/identify 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.

This process is stated below :

* Preprocess the data
* Calculate DTW distance
* Compare DTW distances between buses to find who is following whom.
* Calculate confidence.
* Create a dataframe and store it in a CSV file , who is following whom with the given confidence score.

1. **Background Details**

**2.1 Conceptual Review**

Time series analysis helps organizations understand the underlying causes of trends or systemic patterns over time. Using data visualizations, business users can see seasonal trends and dig deeper into why these trends occur. With modern analytics platforms, these visualizations can go far beyond line graphs.

Deep learning methods offer a lot of promise for time series forecasting, such as the automatic learning of temporal dependence and the automatic handling of temporal structures like trends and seasonality.

In time series analysis, with respect to this project where DTW is used, dynamic time warping (DTW) is an algorithm for measuring similarity between two temporal sequences, which may vary in speed.

The objective of time series comparison methods is to produce a distance metric between two input time series. The similarity or dissimilarity of two-time series is typically calculated by converting the data into vectors and calculating the Euclidean distance between those points in vector space.

In general, DTW is a method that calculates an optimal match between two given sequences (e.g. time series) with certain restrictions and rules.

1. **System Design & Methodology**

**3.1 System Architecture:**

This example illustrates the implementation of the dynamic time warping algorithm when the two sequences *s* and *t* are strings of discrete symbols. For two symbols *x* and *y*, d(x, y) is a distance between the symbols, e.g. d(x, y) = {\displaystyle |x-y|}.

int DTWDistance(s: array [1..n], t: array [1..m]) {

DTW := array [0..n, 0..m]

for i := 0 to n

for j := 0 to m

DTW[i, j] := infinity

DTW[0, 0] := 0

for i := 1 to n

for j := 1 to m

cost := d(s[i], t[j])

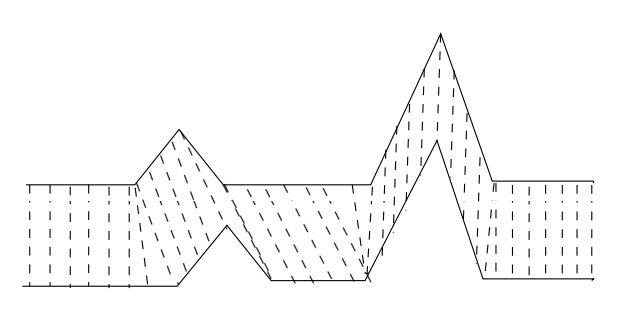
DTW[i, j] := cost + minimum(DTW[i-1, j ], // insertion

DTW[i , j-1], // deletion

DTW[i-1, j-1]) // match

return DTW[n, m]

}



***Figure (1)***

**3.2Development Environment. (H/w & S/W)**

**H/W**

* Device name LAPTOP-1AM5592E
* Processor AMD Ryzen 9 5900HX with Radeon Graphics 3.30 GHz
* Installed RAM 16.0 GB (15.4 GB usable)
* Device ID 6BE3EEBB-7B68-4D37-AE73-XXXXXXXX
* Product ID 00342-42607-66575-XXXXXX
* System type 64-bit operating system, x64-based processor
* Pen and touch No pen or touch input is available for this display

**S/W**

* Virtual Studio Code
* Jypiter NoteBook

**3.3Methodology: Algorithm/Procedures**

**Data**

The data provided by RedBus has 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.

**Summary of the Data: -**

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.

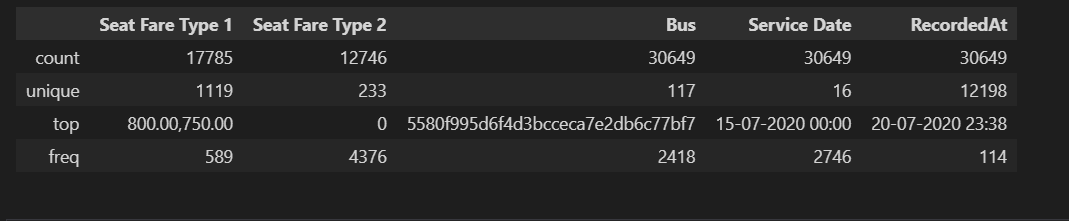
4. Service Date – The date of journey for which the prices are recorded.

5. Recorded At - The time when prices were recorded.

Graphical user interface, text

Description automatically generated

***Figure (2)***

****

## ***Figure (3)***

Graphical user interface, text

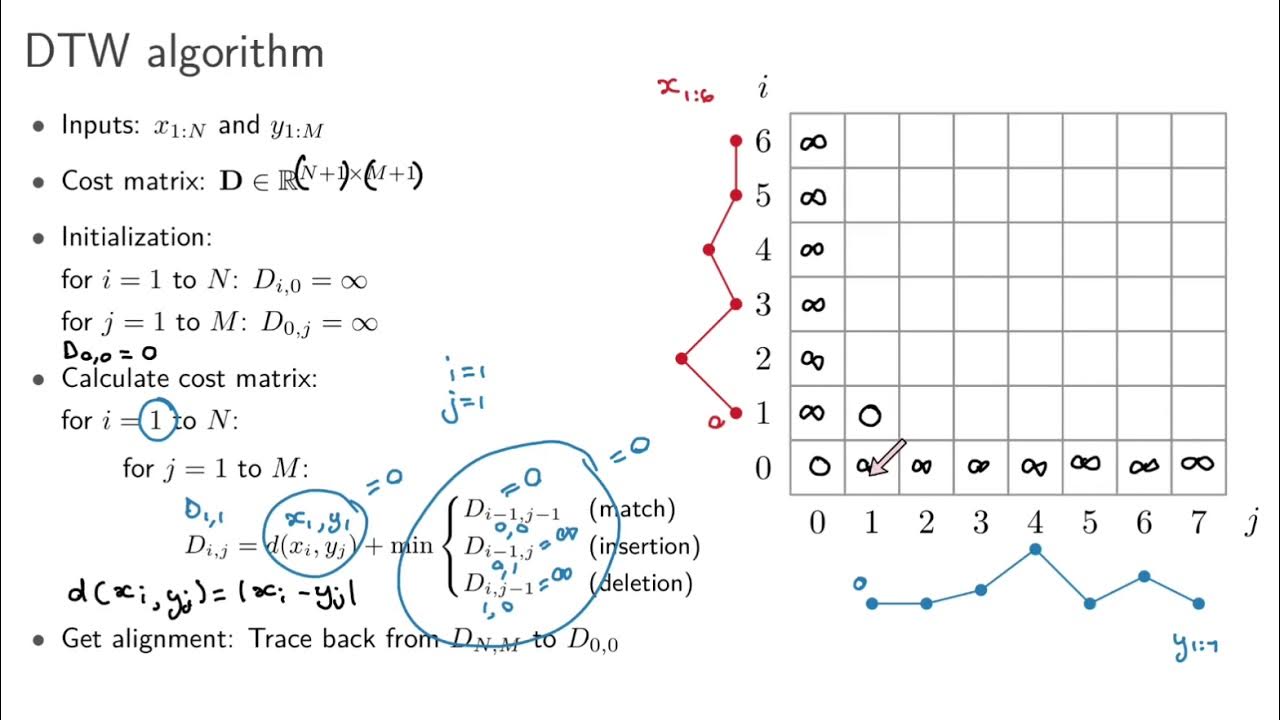
Description automatically generated

## ***Figure (4)***

## **Algorithms Used**

### **Dyanamic Time Wrapping**

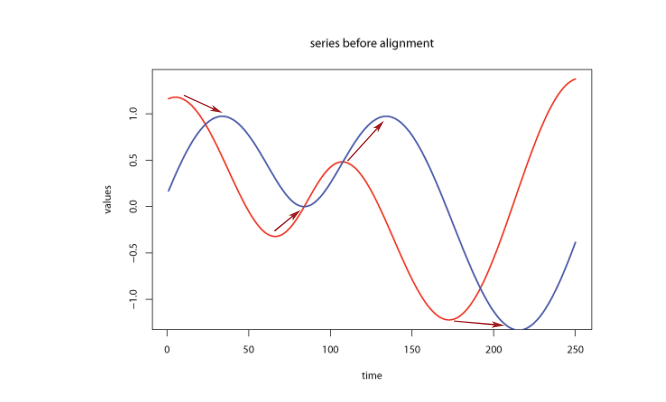
In time series analysis, with respect to this project where DTW is used, dynamic time warping (DTW) is an algorithm for measuring similarity between two temporal sequences, which may vary in speed.



***Figure (5)***

The Model Used in this project: - .

DTW algorithm has earned its popularity by being extremely efficient as the time-series similarity measure which minimizes the effects of shifting and distortion in time by allowing “elastic” transformation of time series in order to detect similar shapes with different phases. Given two time series X = (x1, x2, ...xN ), N ∈ N and Y = (y1, y2, ...yM), M ∈ N represented by the sequences of values (or curves represented by the sequences of vertices) DTW yields optimal solution in the O(MN) time which could be improved 3 Figure 1: Raw time series, arrows show the desirable points of alignment. further through different techniques such as multi-scaling. The only restriction placed on the data sequences is that they should be sampled at equidistant points in time (this problem can be resolved by re-sampling). If sequences are taking values from some feature space Φ than in order to compare two different sequences X, Y ∈ Φ one needs to use the local distance measure which is defined to be a function: d : Φ × Φ → R ≥ 0 (1) Intuitively d has a small value when sequences are similar and large value if they are different. Since the Dynamic Programming algorithm lies in the core of DTW it is common to call this distance function the “cost function” and the task of optimal alignment of the sequences becoming the task of arranging all sequence points by minimizing the cost function (or distance). Algorithm starts by building the distance matrix C ∈ R N×M representing all pairwise distances between X and Y . This distance matrix called the 4 Figure 2: Time series alignment, cost matrix heatmap. local cost matrix for the alignment of two sequences X and Y : Cl ∈ R N×M : ci,j = kxi − yjk , i ∈ [1 : N], j ∈ [1 : M] (2) Once the local cost matrix built, the algorithm finds the alignment path which runs through the low-cost areas - “valleys” on the cost matrix, Figure 2. This alignment path (or warping path, or warping function) defines the correspondence of an element xi ∈ X to yj ∈ Y following the boundary condition which assigne first and last elements of X and Y to each other, Figure 3. Formally speaking, the alignment path built by DTW is a sequence of points p = (p1, p2, ..., pK) with pl = (pi , pj ) ∈ [1 : N] × [1 : M] for l ∈ [1 : K] 5 Figure 3: The optimal warping path aligning time series from the Figure 1. which must satisfy to the following criteria: 1. Boundary condition: p1 = (1, 1) and pK = (N, M). The starting and ending points of the warping path must be the first and the last points of aligned sequences. 2. Monotonicity condition: n1 ≤ n2 ≤ ... ≤ nK and m1 ≤ m2 ≤ ... ≤ mK. This condition preserves the time-ordering of points. 3. Step size condition: this criteria limits the warping path from long jumps (shifts in time) while aligning sequences. While this condition will be discussed in greater details in the Section 3, for now will use the basic step size condition formulated as pl+1 − pl ∈ {(1, 1),(1, 0),(0, 1)}. The cost function associated with a warping path computed with respect to the local cost matrix (which represents all pairwise distances) will be: cp(X, Y ) = X L l=1 c(xnl , yml ) The warping path which has a minimal cost associated with alignment called the optimal warping path. We will call this path P ∗ . By following the optimal warping path definition in order to find one, we need to test every possible warping path between X and Y which could be computationally challenging due to the exponential growth of the number of optimal paths as the lengths of X and Y grow linearly. To overcome this challenge, DTW employs the Dynamic Programming - based algorithm with complexity only O(MN). The Dynamic Programming part of DTW algorithm uses the DTW distance function DTW(X, Y ) = cp ∗ (X, Y ) = min cp(X, Y ), p ∈ P N×M where P N×M is the set of all possible warping paths and builds the accumulated cost matrix or global cost matrix D which defined as follows: 1. First row: D(1, j) = Pj k=1 c(x1, yk), j ∈ [1, M]. 2. First column: D(i, 1) = Pi k=1 c(xk, y1), i ∈ [1, N]. 3. All other elements: D(i, j) = min {D(i − 1, j − 1), D(i − 1, j), D(i, j − 1)}+ c(xi , yj), i ∈ [1, N] , j ∈ [1, M]).



## ***Figure (6)***

**4 Implementation and Results**

**4.1. Modules/Classes of Implemented Project**

* Pandas
* Numpy
* Math
* Dtw
* Dtaidistance from dtw
* Matplotlib
* Datetime

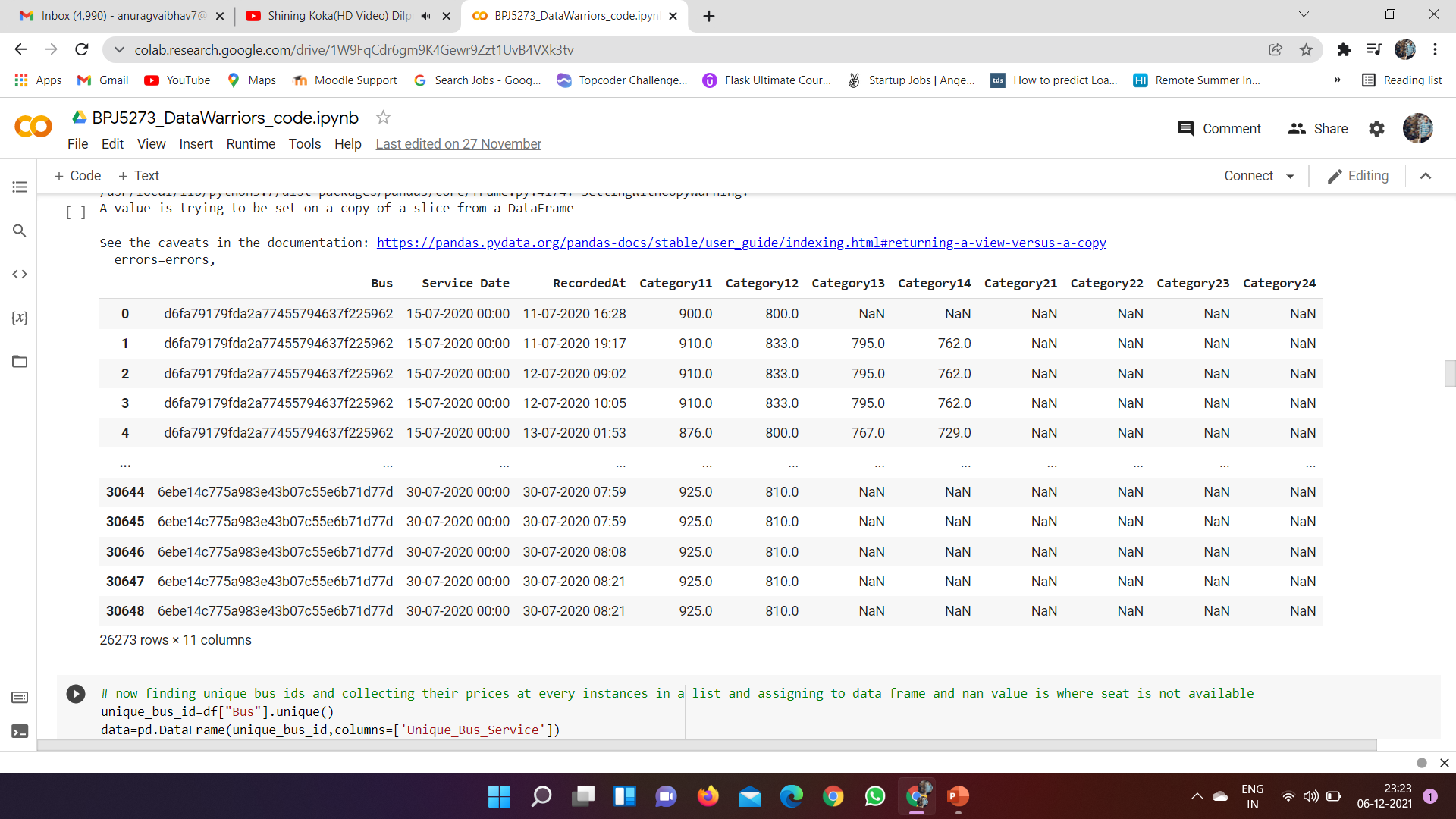
**4.2. Implementation Detail**

Step1 : First, we used pandas for reading the csv dataset file then we performed a data cleaning operation to remove rows having ‘Seat Fare Type 2’ value zero.

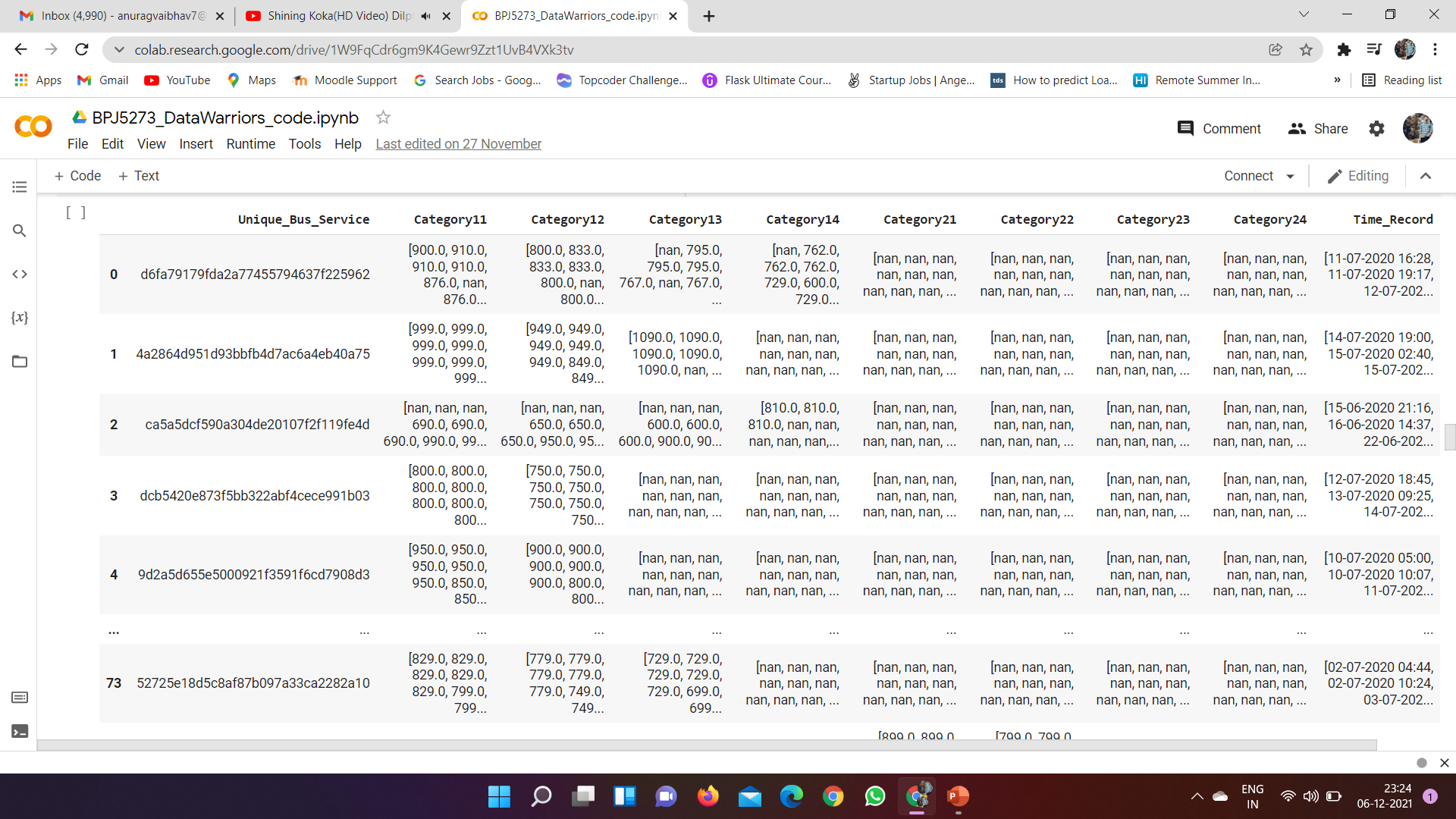
Step2 : In the next step, we made the ‘Seat Fare Type1’ and ‘Seat Fare Type 2’ split into 4 categories each depending on subcategories(front/back/upper/lower) in this process we created 8 new columns like Category11 for Seat Fare Type 1 front seat, Category12 and so on.

Step3 :This step is a bit complicated to understand but to summarise it. We first used a unique method to find several unique buses in the dataset. Later we iterated over the dataset and we started storing all the prices and time of Category11 for a bus in a list . Then we created a new data frame in which we stored unique buses and corresponding to it we stored a list of prices along with the time in the list.

Represntation of the above three steps:



***Figure (7)***



Step4 : In this step, we have sorted the time list for each bus and accordingly we sorted the price list and converted the date in Date Time format then we started plotting time-series graphs for each category column.

Step5: We found that the time series graph of Category21, Category22 ,Category23 have only 1 or 2 bus data so we decide to drop it off.

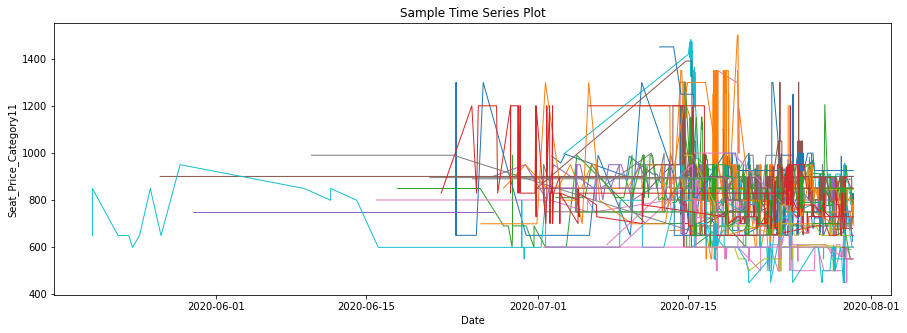
Step6 : In this step, we used a dynamic time wrapping algorithm to compare time series. We calculated the dtw distance between buses for each category and stored it in a list and if unfortunately, we got dtw distance as infinity then we stored -1 in place of it and then later replaced it with the max of that list +1000 to prevent our analysis to produce poor results.

Step7 : Now we added dtw distance of each category together and obtained a compared distance list for each bus. Then we used minimum distance in the list which corresponds to the closest comparison. Using this we're able to find who follows whom then we normalized the distance using (M-S(x,y))/M and then passed it to the sigmoid function to get a confidence score.

Step 8 : Finally we have saved it in CSV file.

#### **4.3 Results and Discussion**

#### Time Series Graph for Each category of Bus\_type



***Figure (8)***

Chart

Description automatically generated

***Figure (9)***

Chart

Description automatically generated

***Figure (10)***

Chart, line chart, histogram

Description automatically generated

***Figure (11)***

Chart, line chart

Description automatically generated

***Figure (12)***

Chart, line chart

Description automatically generated

***Figure (13)***

Chart, line chart

Description automatically generated

***Figure (14)***

Chart, line chart

Description automatically generated

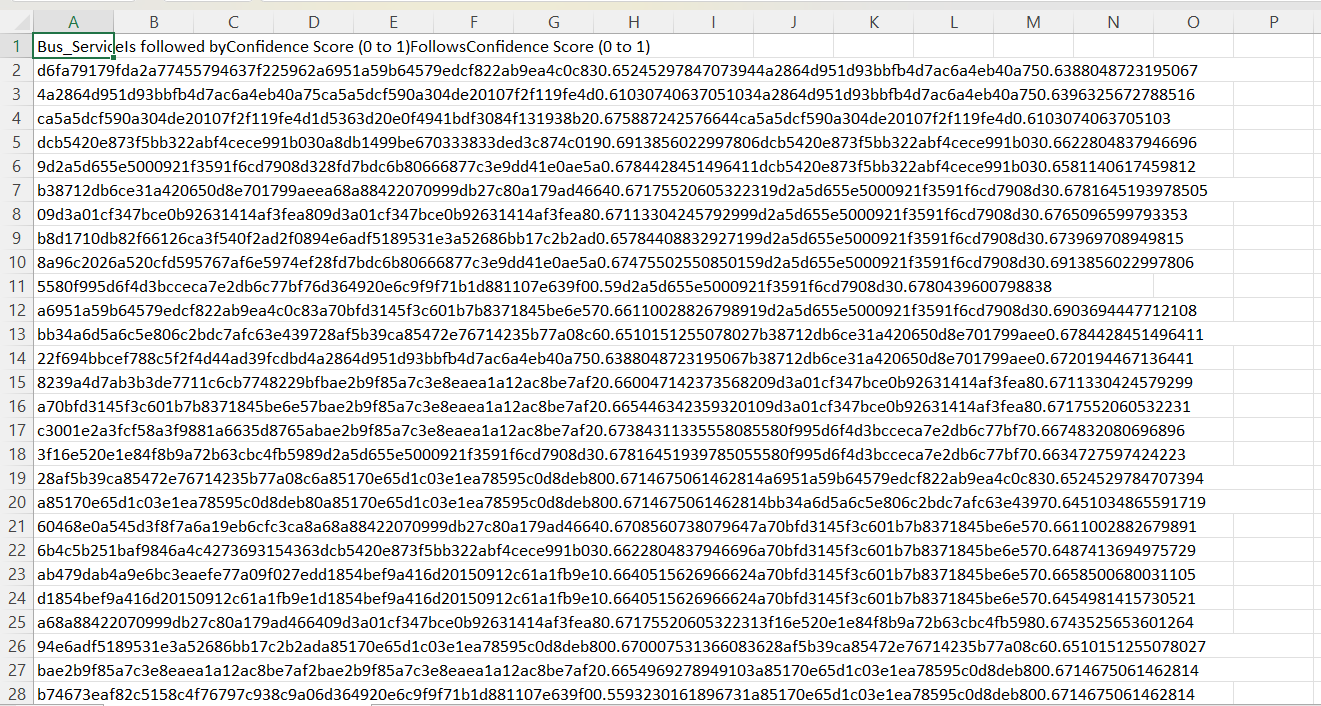
***Figure (15)***

The Confidence score for each Bus\_Service

Text

Description automatically generated

The Final output of the project in a CSV file format containing who is following whom and confidence score for each such instance.



**CONCLUSION**

In this project the data set given by RedBus was used to is to find out which services is/are pricing independently (the priceleaders) and which services are the followers (and following whom?), given the history of prices set by the operators.

All the background preprocessing was done to restructure the data set and it was given to the Dynamic Time Wrapping Alogoritm to calculate Dtw distances and finally conclusion was made who is following whom.

To conclude we can state that we have proposed a model that can reveal the market dynamics, which usually occurs in the price-setting game for a bus. This method should broadly classify which bus operator changes price based on supply and demand independently and who is a follower of other price setters. We have also produced a confidence score in favor of what we predicted.

Finally solving the problem for the operators who does not have the same level of visibility on these signals of demand and supply or the capability to price effectively.

**References**

1. Pavel Senin, “Dynamic Time Warping Algorithm Review”, ALAOUI OUATIK(2014),A , 1, 2014, 16