A logo for college computing

Description automatically generated**Assessment Cover Page**

**Declaration**

By submitting this assessment, I confirm that I have read the CCT policy on academic misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source.

I declare it to be my own work and that all material from third parties has been appropriately referenced.

I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

|  |  |
| --- | --- |
| *Student Full Name: Caroline Floss Bresolin* |  |
| *Student Number: 2023190* |  |
| *Module Title: Machine Learning for Business* |  |
| *Assessment Title: CA1 Project* |  |
| *Assessment Due Date: 21/04/2024* |  |
| *Date of Submission: 19/04/2024*  **Use of AI Tools**  I have not used any AI tools or technologies in the preparation of this assessment. |  |

[Introduction 3](#_Toc163236632)

[Data understanding 3](#_Toc163236633)

[Clustering 5](#_Toc163236634)

[KMeans 8](#_Toc163236635)

[Hierarchical Agglomerative Cluster 12](#_Toc163236636)

[Summary Cluster Models 15](#_Toc163236637)

[ARIMA 16](#_Toc163236638)

[*Problem Description and Dataset* 16](#_Toc163236639)

[*Data Analysis* 17](#_Toc163236640)

[*Order Differencing* 20](#_Toc163236641)

[Model parameters 21](#_Toc163236642)

[Auto ARIMA 22](#_Toc163236643)

[Metrics Auto Arima: 23](#_Toc163236644)

[Residuals Auto Arima 24](#_Toc163236645)

[ARIMA and Walk-forward validation 25](#_Toc163236646)

[Metrics with Walk-forward validation 27](#_Toc163236647)

[Optimal parameters 28](#_Toc163236648)

[Reference List 29](#_Toc163236649)

# Introduction

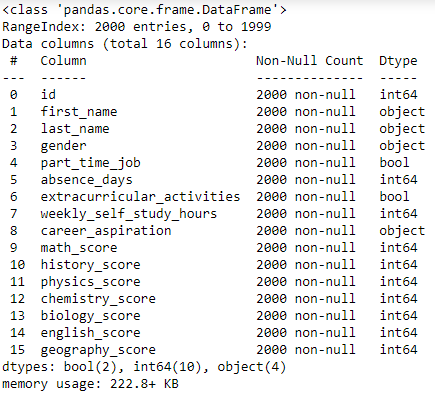
We were tasked with developing and deploying machine learning models using a dataset related to one of the listed subjects. The dataset was sourced from Kaggle, which contains information on the academic performance of senior students in a fictional high school at the end of their final semester (www.kaggle.com, n.d.).

In the realm of education, clustering analysis serves as a potent tool for uncovering patterns within student performance data. By clustering students based on similarities, we can effectively tailor interventions and enhance educational outcomes.

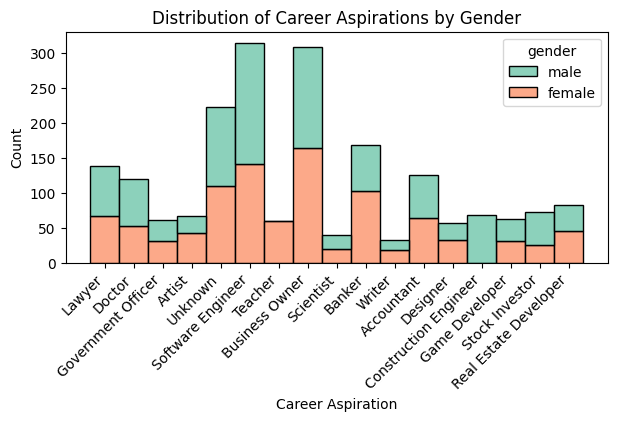
My clustering efforts aim to categorize students into meaningful groups, focusing on academic performance and weekly study hours. The dataset provides pertinent and easily accessible information, making it well-suited for clustering analysis.

# Data understanding

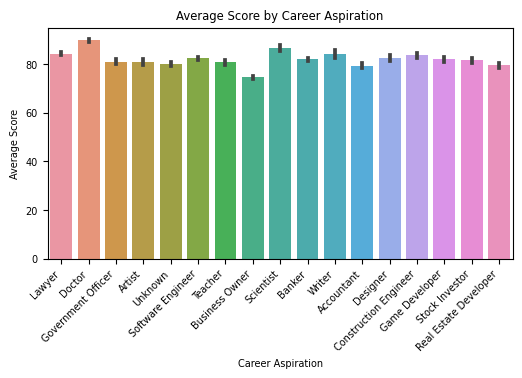
After inspecting the dataset with standard statistical functions, EDA is employed to get more understanding of it.



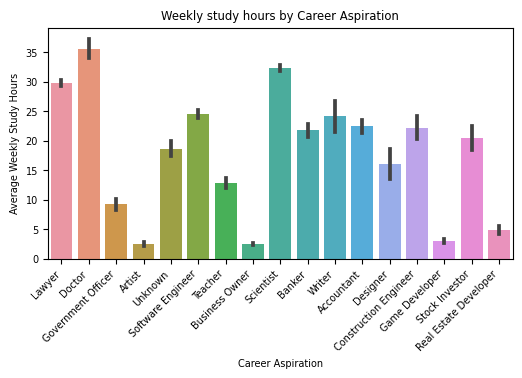
The image below shows that most career aspirations are balanced by gender, except “Teacher” and “Banker” being more representative by females, and Construction Engineer by males.



The average score is calculated based on the seven lecture scores contained in the dataset. With that, we can analyse the average score by career aspiration. It suggests that those who possess the highest average scores are aspiring to become doctors, on the other hand with the lowest average scores are those who are aspiring to become a Business Owner.



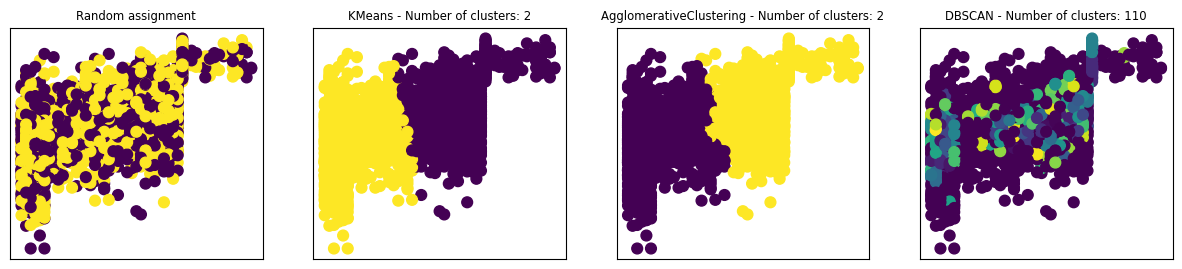
When analyzing the weekly study hours based on career aspirations, it's noteworthy that individuals who dedicate more time to studying tend to achieve higher average scores.



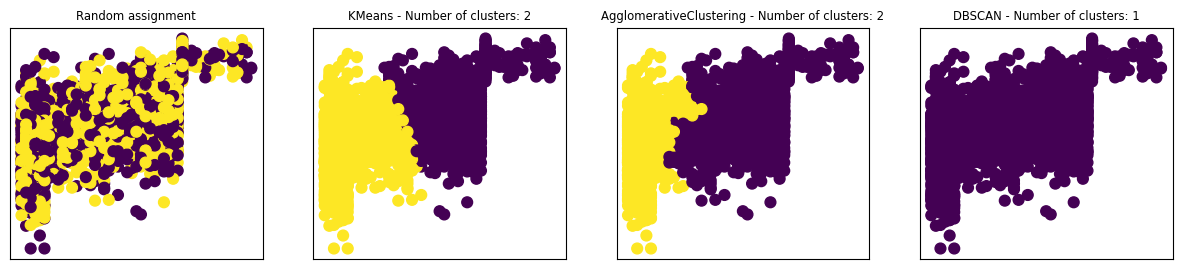
# **Clustering**

There are several methods to perform clustering and deciding which is more beneficial depends on the characteristics of the clusters, the features of the dataset, the number of outliers, and the number of data objects. Before going deeper into the chosen methods, we can observe how this dataset behaves among the three cluster models KMeans, Agglomerative, and DBSCAN.

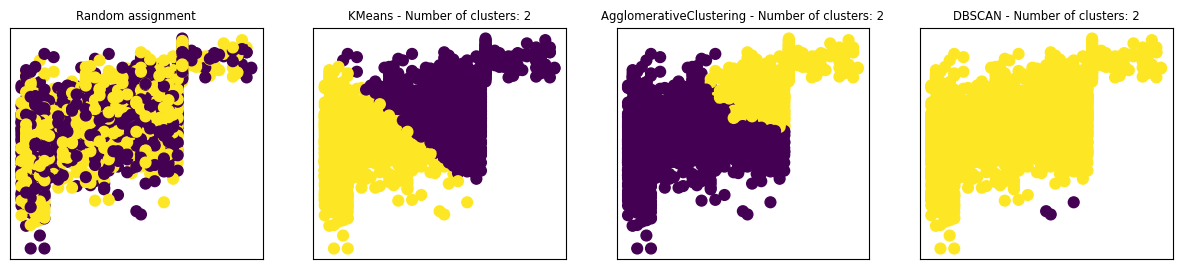
Without applying any scaling method:



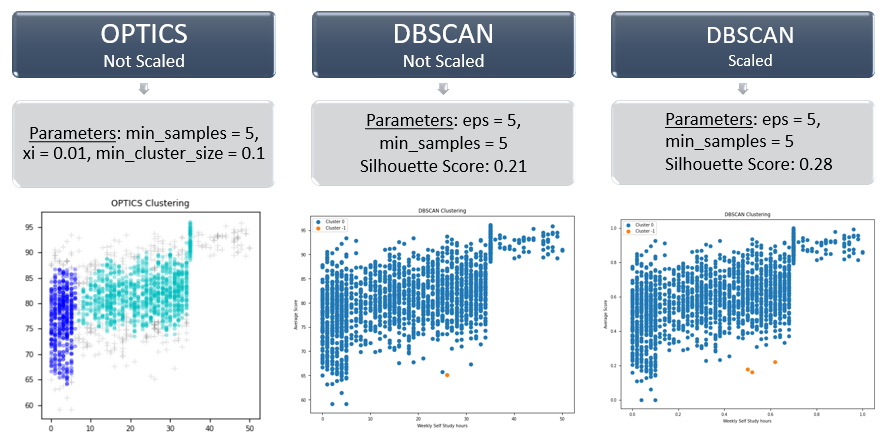
MinMaxScaler:

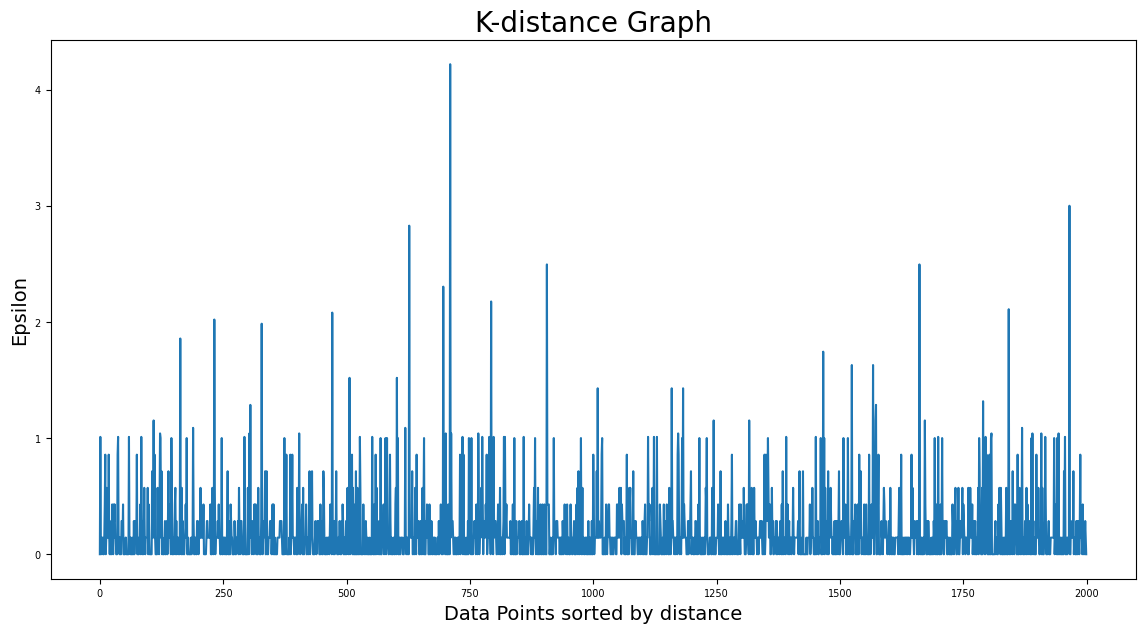


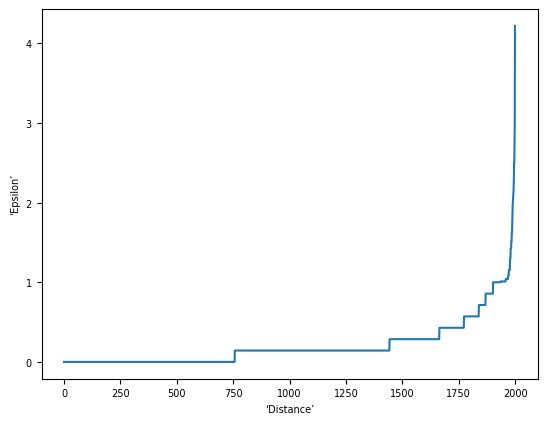
Standard Scaler:



OPTICS and DBSCAN algorithms failed to demonstrate effective clustering, even after adjusting the parameters:





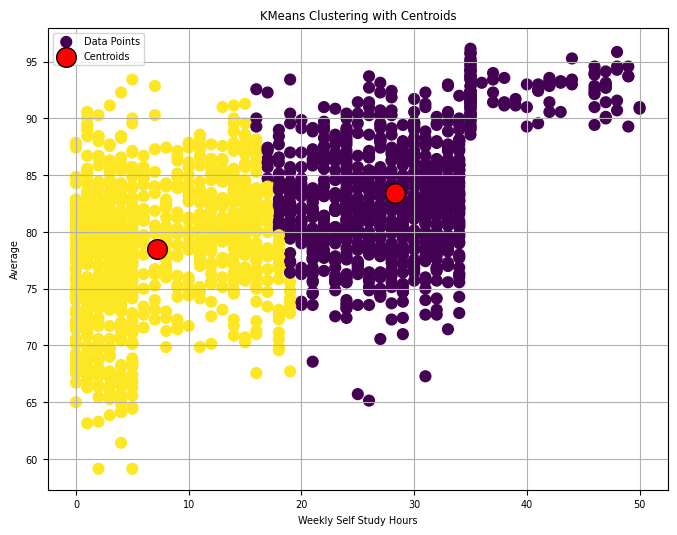


Based on the distribution of the selected features, KMeans, and Agglomerative Hierarchical were the two that demonstrated the best approach.

## KMeans

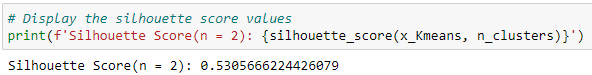
This algorithm iteratively divides data points into K clusters by minimizing the variance in each cluster. First, each data point is randomly assigned to one of the K clusters. Then, it computes the centroid (functionally the center) of each cluster and reassigns each data point to the cluster with the closest centroid. This process repeats until the cluster assignments for each data point are no longer changing (www.w3schools.com, n.d.).

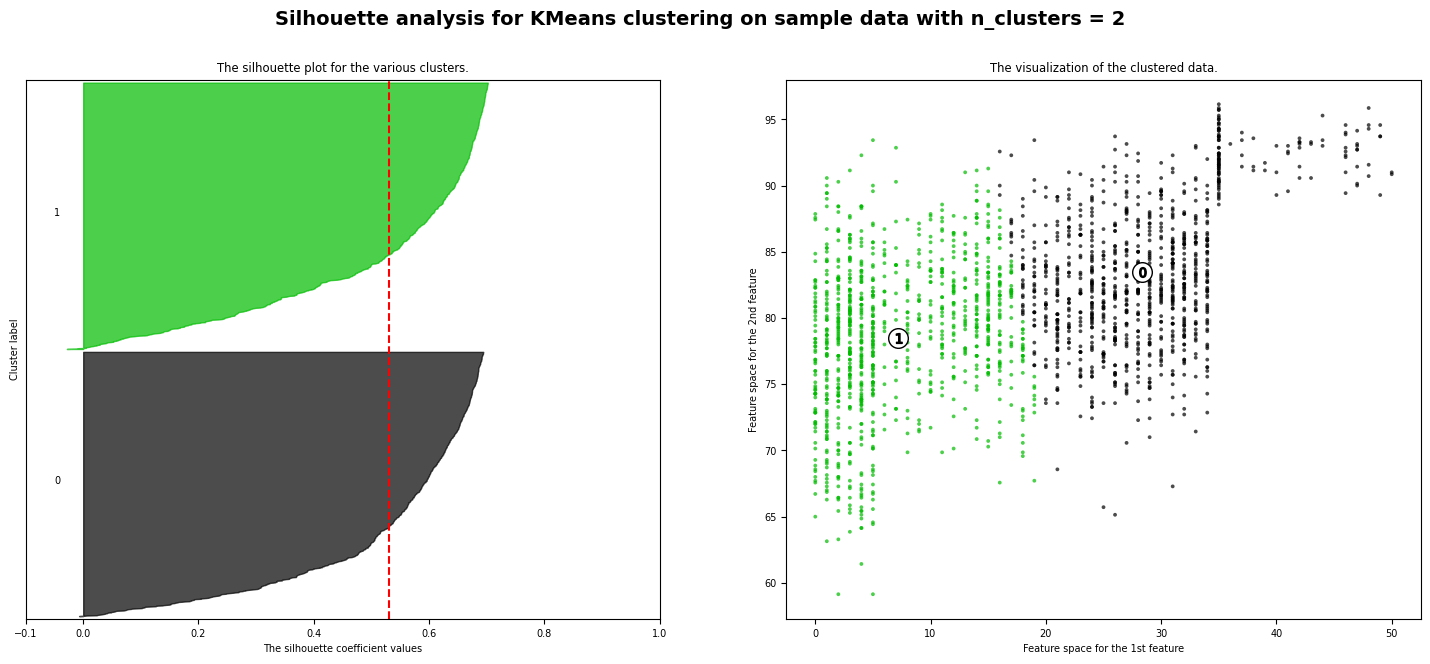
Below we can visualize it divided into two clusters:

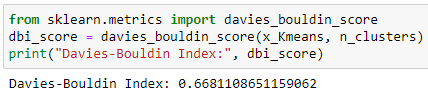


To measure how well the model performed, we can employ the Silhouette Score. It measures how similar an object is to its cluster compared to other clusters. A score closer to 1 indicates that the object is well-matched to its cluster and poorly matched to neighboring clusters.

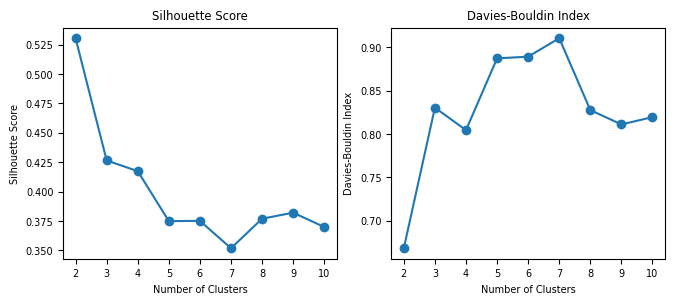
The Silhouette score yields a reasonable outcome, attributed to the distribution of the data points. For instance, in the first cluster, there are fewer students with low averages and fewer with high averages increasing the distance between the points. Furthermore, the clusters are not distinctly separated, resulting in nearby points belonging to different clusters.



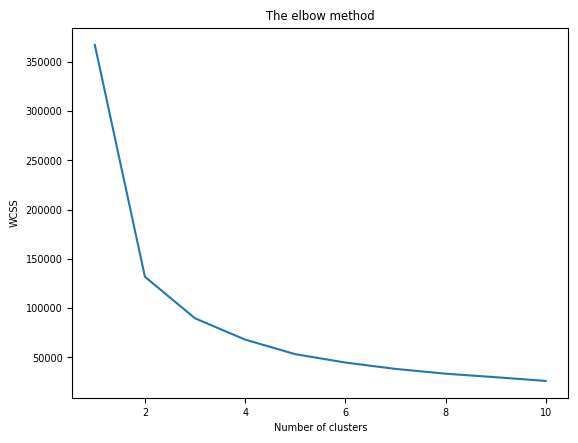


The Davies-Bouldin Index evaluates the average similarity between clusters, where a lower value signifies improved separation. At 0.67, the clusters exhibit reasonably good separation. However, certain data points in the middle display similar distances to both centroids, leading to less distinct cluster boundaries. This phenomenon is attributable to the dataset's distribution.

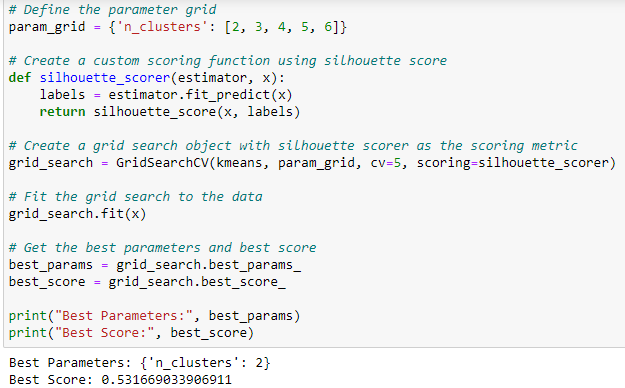
Both methods yield the best performance with two clusters:



The elbow method is a common approach to determining the optimal number of “k”



Another way is using GridSearchCV, which was the one employed considering its metrics and outcomes:

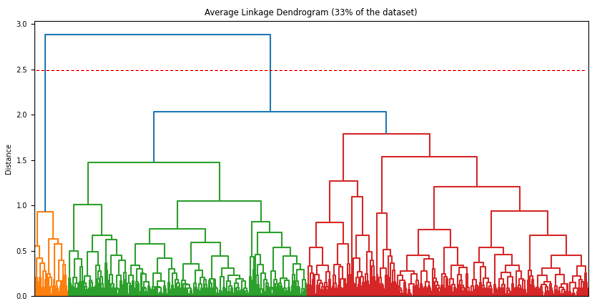


## Hierarchical Agglomerative Cluster

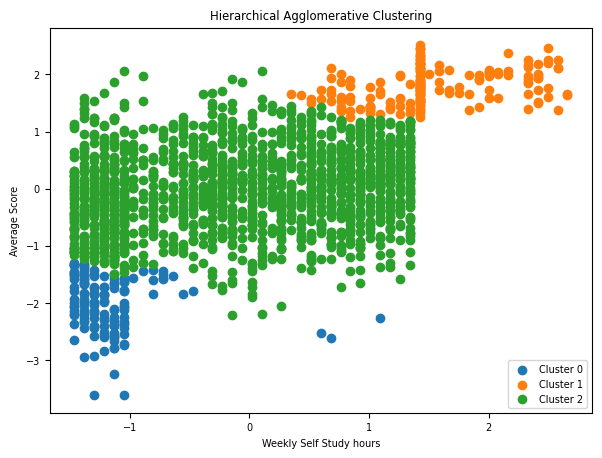
Hierarchical clustering is an unsupervised learning technique used to group similar objects into clusters. It creates a hierarchy of clusters by merging or splitting them based on similarity measures (Sharma, 2019).

In hierarchical clustering, we use distance metrics like Euclidean and Manhattan to assess cluster similarity. Euclidean measures straight-line distance, which is the one used here, while Manhattan considers horizontal and vertical steps.

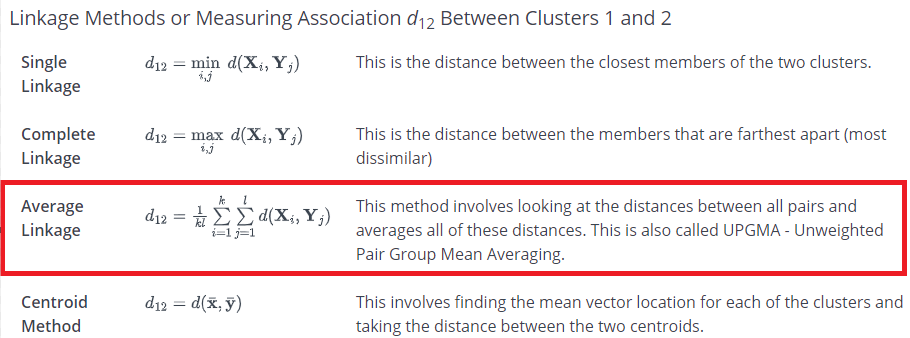
To determine cluster number, we can use a dendrogram, showing cluster hierarchy and natural groupings. Setting a threshold distance, we draw a line (often at the tallest vertical line). In the dendrogram below, a line at 25 yields 2 clusters, one smaller.



If we utilize 3 clusters as initially separated, their results would not be beneficial for this proposal. The image below illustrates their appearance:

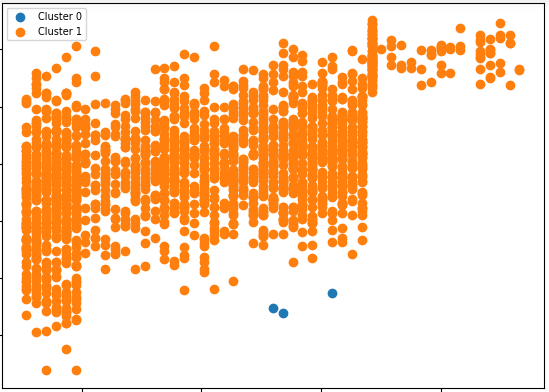
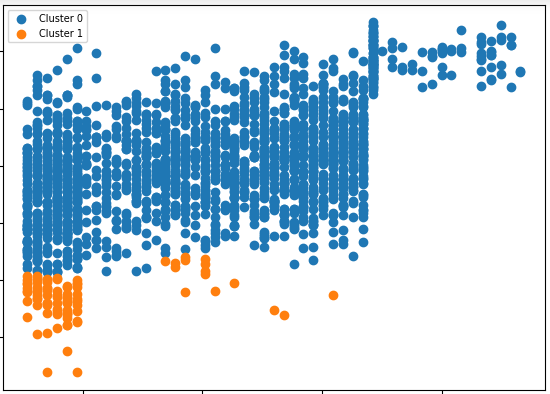


As a “linkage” parameter the average is used, it is determined depending on the purpose of the clustering. Average linkage determines the distance between two clusters as the average distance between all pairs of points in the two clusters. This method tends to produce clusters that are somewhere between the long, chain-like clusters produced by single linkage and the compact, spherical clusters produced by complete linkage (Bismi, 2023).

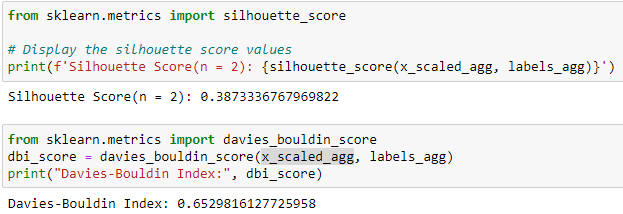
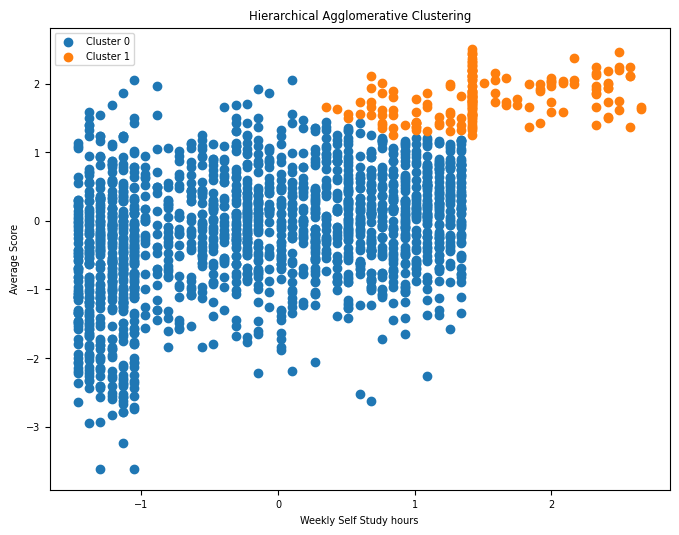


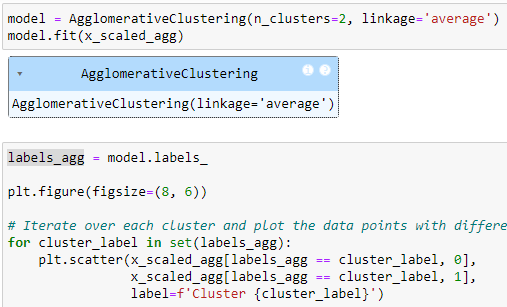
(PennState: Statistics Online Courses, n.d.)

Single method at the image below on the right and Complete linkage at the left.

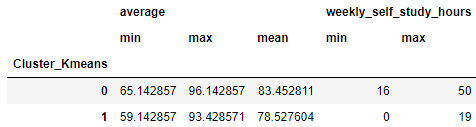


With two clusters we achieved the highest silhouette score and can group the students according to their performance and their dedicated self-study weekly hours.

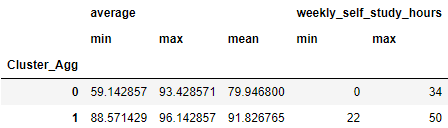




### Summary Cluster Models

KMeans primarily grouped students based on their weekly self-study hours, resulting in similar average scores across clusters. This outcome is attributed to the algorithm's process of reassigning points based on their proximity to centroids.

On the other hand, Hierarchical Agglomerative clustering, by creating a hierarchy of clusters through merging based on similarity measures, proved more effective for our project's purpose. It successfully separated students based on their academic performance, considering both average scores and weekly self-study hours. Specifically, it highlighted students who had the highest average scores and spent more time studying from those with low or high average scores but less study time.



In summary, while the model provided valuable insights, there is room for improvement in better distinguishing between clusters.

# ARIMA

ARIMA is a general class of statistical models for time series analysis forecasting. It stands for Auto-Regressive Integrated Moving Average. When applying ARIMA models, we use a time series of past values and/or forecast errors to predict future values (Justin, 2022).

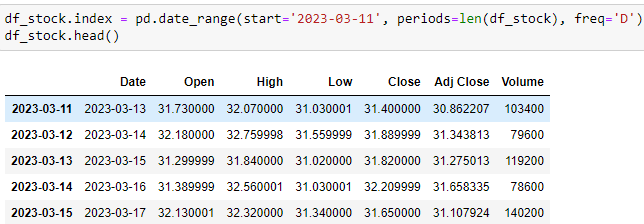
## *Problem Description and Dataset*

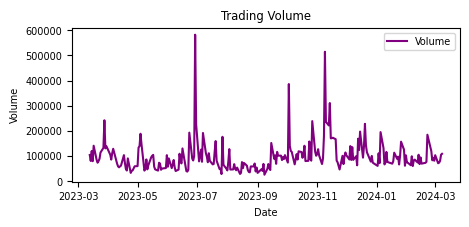
The issue was forecasting the daily Close stock prices for Carriage Services. The dataset utilized contains the daily stock prices spanning from 13 March 2023 to 8 March 2024 and was sourced from Yahoo Finance (finance.yahoo.com, n.d.).

## *Data Analysis*

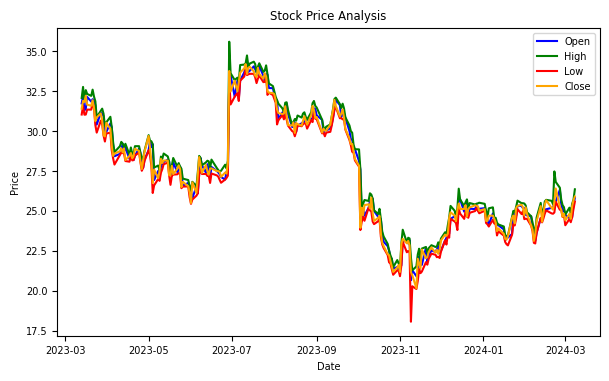


The column “Date” is adjusted to date format:

**

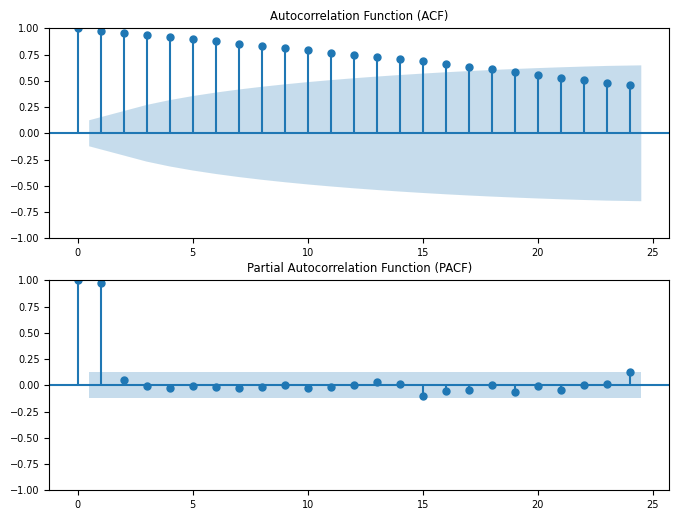


*Figure 1*

**

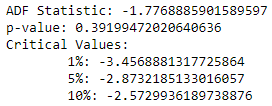
*Figure 2*

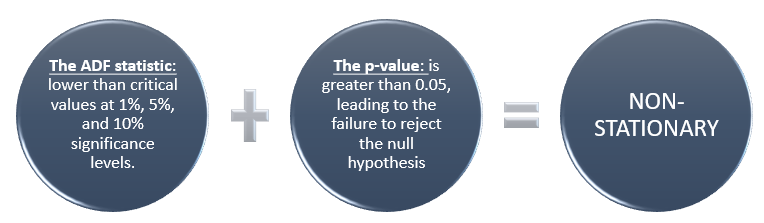
Stationarity requires that statistical properties, like mean, variance, and covariance, stay constant over time. However, in this case, the mean of the 'Close' feature fluctuates over time. Moreover, the ACF plot shows high and positively correlated lags with decay, while the PACF plot displays a single spike at lag 1, both indicating a trended time series. Therefore, our time series is non-stationary.

**

*Figure 3*

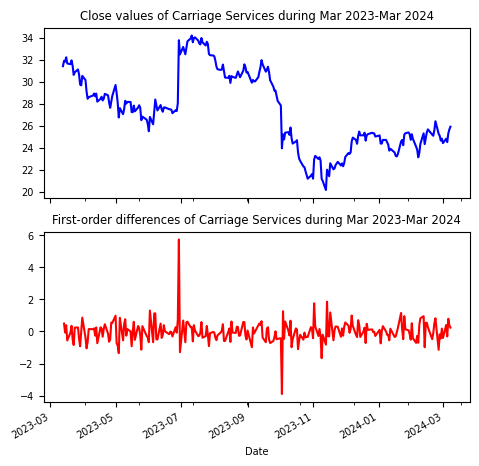
Besides plots, we can also check for stationarity with statistical tests like the ADF. It tests for the null hypothesis that there is a unit root (non-stationary).



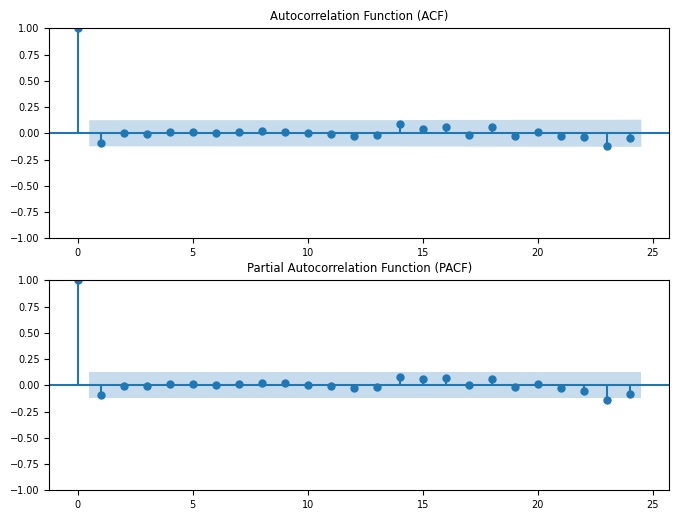


## *Order Differencing*

Due to non-stationarity in the dataset, differencing can be applied to make it stationary. Comparing the transformed series with the original reveals a reduced trend in the first difference. However, even after differencing, both autocorrelation and partial autocorrelation functions show similarities, possibly indicating remaining autocorrelation due to factors like seasonality. While ACF and PACF plots provide valuable diagnostic insights for ARIMA modelling, overall model performance should be evaluated using various metrics, including forecast accuracy and statistical significance. A successful model captures essential data features effectively, even if diagnostic plots don't perfectly match expectations.



*Figure 4*

**

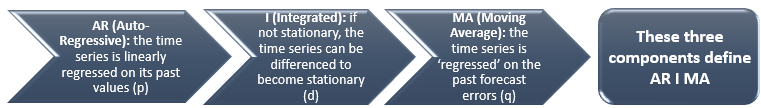
*Figure 5*

We can also quantify the result by running the ADF test on the series.



The p-value is enough to reject the null hypothesis. We can conclude that our first difference time series will likely be stationary. Said that our (d)parameter is settled to 1.

## Model parameters

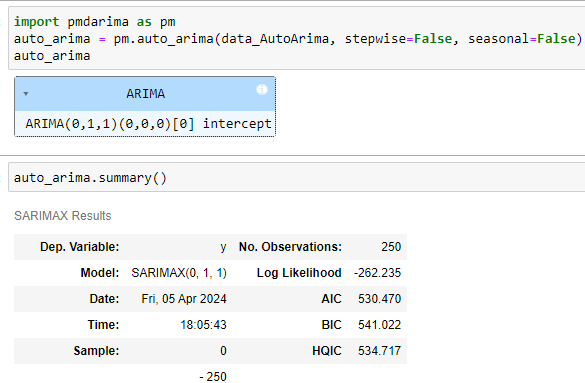


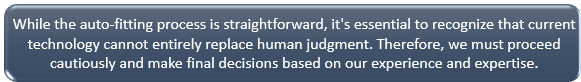
Once the (d) parameter is determined, we proceed to define the remaining two parameters, p and q. Utilizing the ACF and PACF plots on the first difference time series, we can determine initial values for p and q. Both plots reveal insignificant lags, resulting in an ARIMA (0,1,0) model, also referred to as a white noise model.

In the stock price context, lacking significant autocorrelation implies changes driven by the random arrival of new information. This aligns with the efficient market hypothesis, where prices rapidly reflect available information. Once parameters are set, we construct and assess the model's metrics.

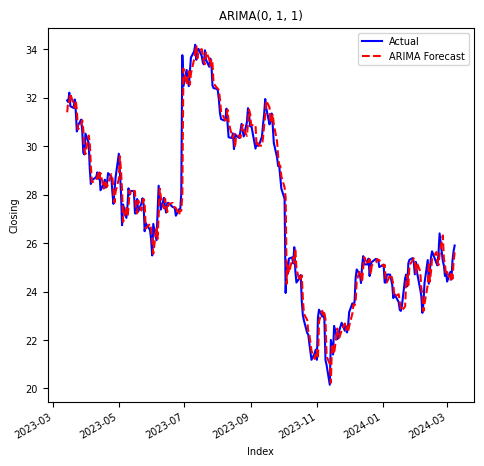
# Auto ARIMA

The *pmdarima* Python package offers automatic ARIMA modelling based on the *stats models* library. It will generate the optimal model based on its criteria.

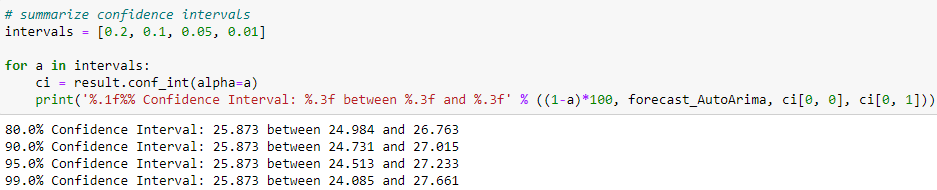


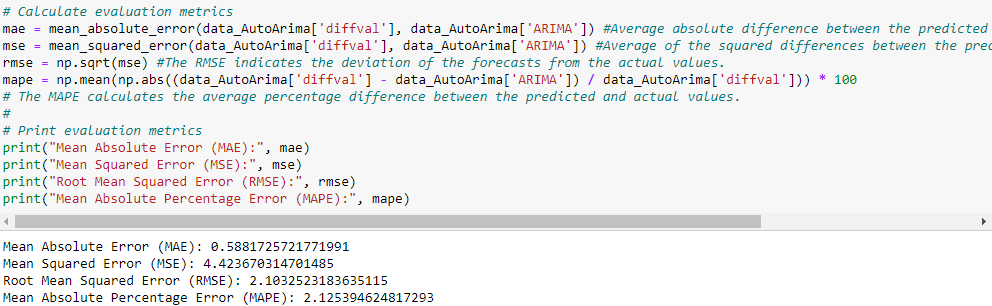


The image below shows forecasted values and the actual values



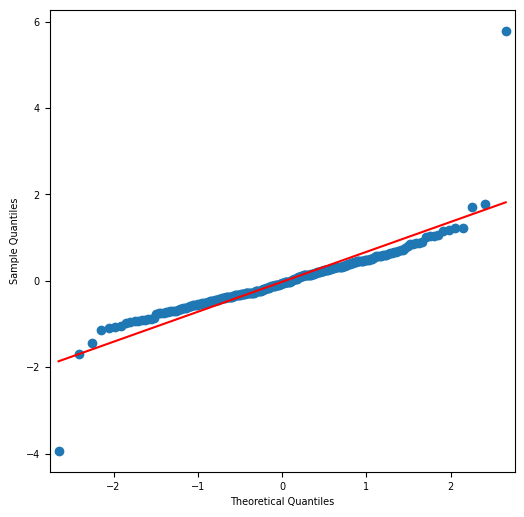
### Metrics Auto Arima:

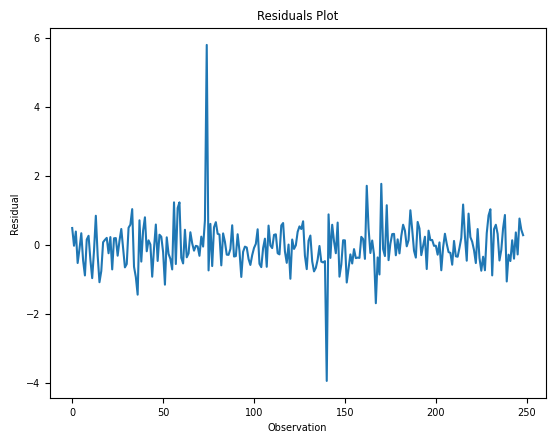




### Residuals Auto Arima

They are the difference between an observation and its predicted value at each time step.



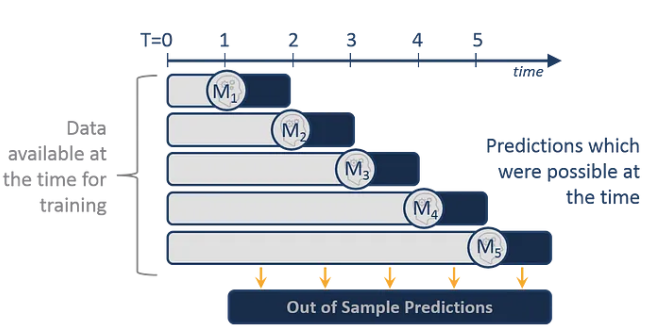


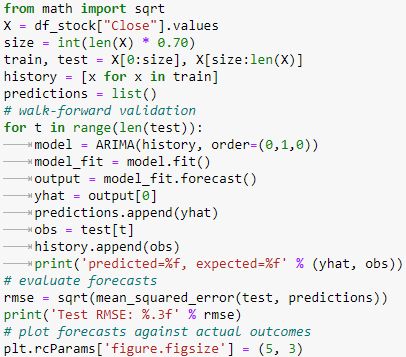
The residuals are interesting, showing periods of high variability particularly towards the end of May and July. To mitigate this variability, one approach is to employ cross-validation. For this purpose, walk-forward validation is employed in the following steps.

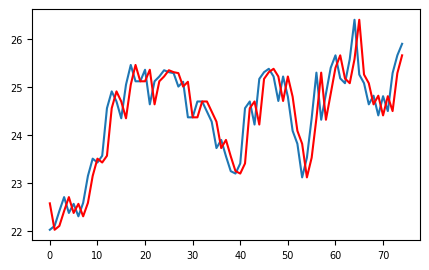
## ARIMA and Walk-forward validation

It realistically assesses performance, reflecting real-world scenarios with new data over time. It helps detect overfitting or underfitting, aiding necessary adjustments.

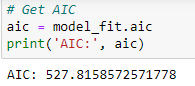
In time series modeling, the predictions over time become less and less accurate and hence it is a more realistic approach to re-train the model with actual data as it becomes available for further predictions. Since training of statistical models is not time-consuming, walk-forward validation is the most preferred solution to get the most accurate results (Tripathi, 2021).

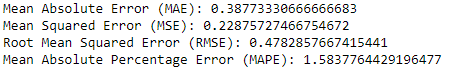


The parameters are set as initially identified (0,1,0):

**

### Metrics with Walk-forward validation

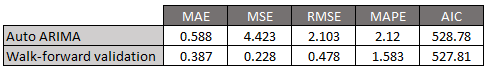




The MAPE value indicates that, on average, the model's forecasts deviate from the actual values by about 1.58%. RMSE values with different training sets do not necessarily lead to better performance.

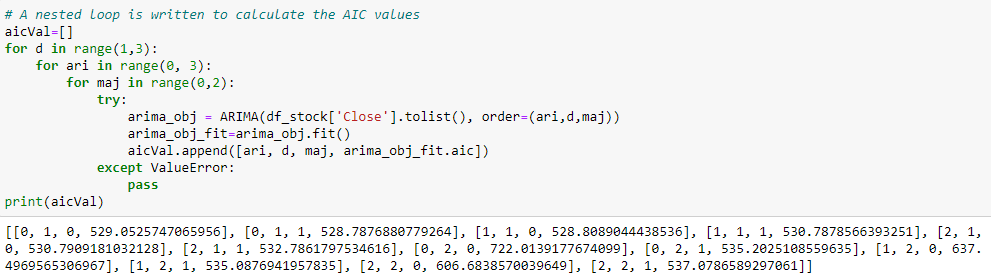


It shows that employing cross-validation can potentially improve the metrics.



## Optimal parameters

It suggests that the best parameters are (0,1) with an AIC of 528.78. However, a better result was achieved with (0,0), resulting in an AIC of 527.81. The RMSE remains similar.



Counting words:



# Reference List

Bismi, I. (2023). *Different Linkage Methods used in Hierarchical Clustering*. [online] Medium. Available at: https://medium.com/@iqra.bismi/different-linkage-methods-used-in-hierarchical-clustering-627bde3787e8.

PennState: Statistics Online Courses. (n.d.). *14.4 - Agglomerative Hierarchical Clustering | STAT 505*. [online] Available at: https://online.stat.psu.edu/stat505/lesson/14/14.4.

Sharma, P. (2019). *Hierarchical Clustering | Hierarchical Clustering Python*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2019/05/beginners-guide-hierarchical-clustering/.

Tripathi, H. (2021). *Walk through TIME — Part-2 (Modelling of Time Series Analysis in Python)*. [online] Medium. Available at: https://tripathitesh.medium.com/walk-through-time-part-2-modelling-of-time-series-analysis-in-python-eb1f077c4d06 [Accessed 26 Mar. 2024].

www.w3schools.com. (n.d.). *Python Machine Learning - K-means*. [online] Available at: https://www.w3schools.com/python/python\_ml\_k-means.asp.