cct College Dublin
Computing • IT • Business

Assessment Cover Page

Student Full Name	Izaias de Oliveira Gomes Junior
Student Number	2023232
Module Title	Machine Learning for Business
Assessment Title	CA1 Project
Assessment Due Date	23rd April 2024
Date of Submission	23rd April 2024

Use of AI Tools

I acknowledge the use of **ChatGPT and Gemini** for the purpose of **[provide a brief explanation of how you used the tool].**

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.

Contents

Introduction

This project will deal with two different Machine Learning techniques: Clustering and Time Series. Two datasets will be used to perform the techniques.

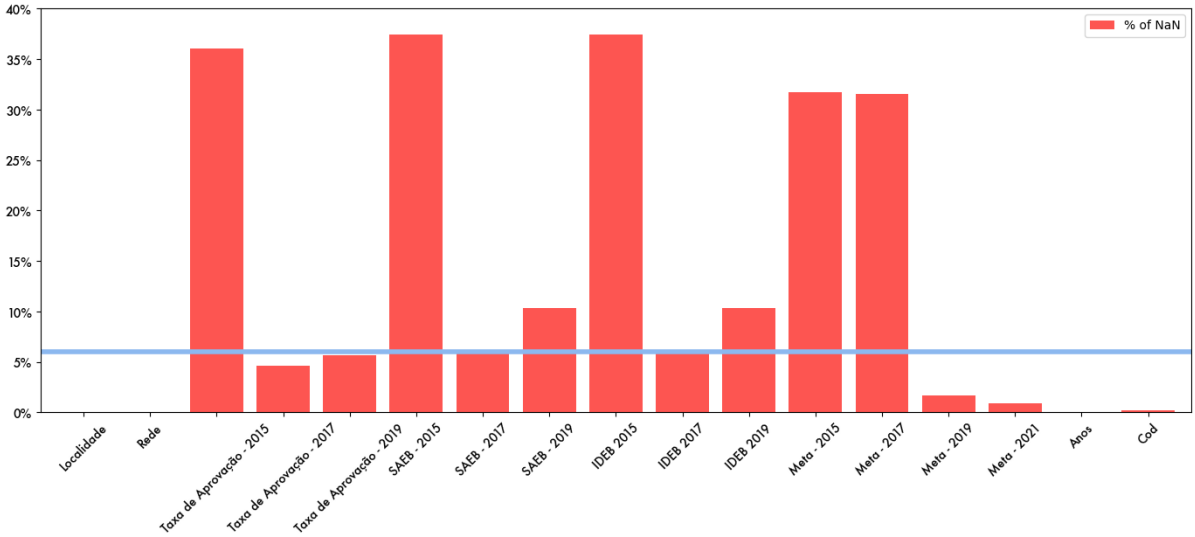
Education is the theme chosen to work with clustering. The dataset contains information about que quality and approval rate of schools in the Brazilian state of São Paulo. Every two years the Brazilian Federal government evaluates the quality of education in city and state schools through the SAEB system ("Basic Education Assessment System" in a free translation). The idea is to assess the knowledge of the students in two basic subjects, Portuguese and Mathematics, and elaborate and monitor educational policies (Governo Brasileiro, 2023). The clustering technique will help to identify if there is any major difference between the city and the state schools.

For Time Series the dataset was retrieved from the Yahoo Finances website, the chosen stock was from the Brazilian pharmaceutical Raia Drogasil SA. The company has more than 1,800 retail stores across Brazil, with a diverse range of medication, cosmetics and other health and beauty products (Infomoney, 2024).

Clustering

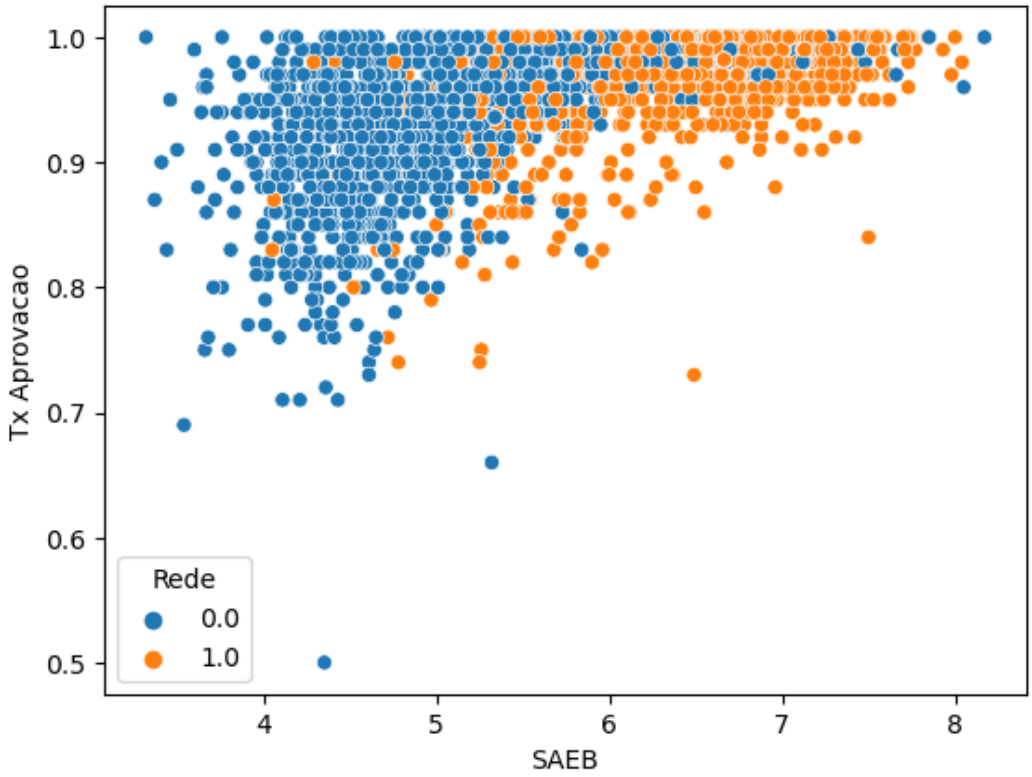
The dataset was retrieved on the statistical agency of the São Paulo state website (SEADE) and contains the quality indicator (SAEB) and the approval rate for the years 2015, 2017 and 2019. Only the year of 2017 will be used due the low amount of missing values (roughly around 6%) compared to the other two years (Plot 1).

PLOT 1: Proportion of Missing Values



Plot 2 shows the relationship between Approval Rate (y axis) and Quality Rate (x axis) with the blue spots representing the state schools and the orange spots the city schools. Before applying any clustering method it is clear that the data points are divided into two groups: the majority of state schools have a SAEB rate less than 6 and approval rate between 0.7 and 1.0; and the city schools have a higher quality rate (SAEB higher than 6) with the approval rate ranging between 0.9 and 1.

PLOT 2: Scatterplot: Approval and Quality Rate by type os School

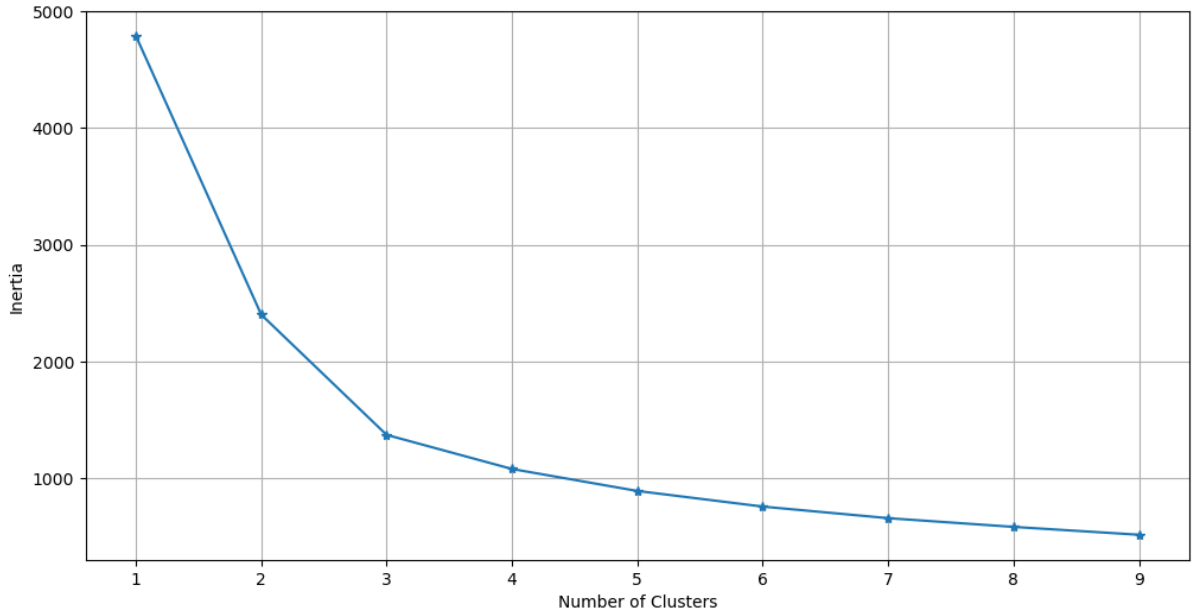


Clustering: K-Means and K-Medoids

After the standardisation of the data using the Standard Scaler from Sklearn, the Elbow Method was plotted to decide the ideal number of clusters for the first algorithm: K-Means. This step is important because different scales in features might cause problems on the algorithm due its distance-based calculations.

The Elbow Method is a graphical representation that helps to find the ideal number of clusters to apply the K-Means algorithm. The WCSS is the sum of the square distance between the data points in a cluster and its centroid. The number of clusters is chosen when the shape of the curve is similar to an elbow; from this point forward the within-cluster sum of squares (WCSS) does not reduce significantly with the increase of the number of clusters (Tomar, 2023). Other methods and the domain knowledge have to be accounted for before the final decision.

PLOT 3: Elbow Method



K-Means is an unsupervised Machine Learning algorithm which separates the observations in a dataset based on the similarity of the points. The groups, or the clusters, are defined based on their centroids and the distance between the other observations (Piech, 2013). The K-Means works with the minimisation of sum of the squared distances between the centroids and the other points within the clusters (Sharma, 2019).

Two more methods that help to decide the number of clusters is the Silhouette Coefficient, which measures the similarity of the data points within the designated clusters (Banerji, 2021); and the Davies-Bouldin Index, that takes into consideration inter and intra clusters distances, a index closer to 0 indicates that the clusters are separated perfectly (Ros, Riad and Guillaume, 2023).

On Table 1 the results for both methods signal that the ideal number of clusters using K-Means is 3, with the highest Silhouette Score and lower Davies-Bouldin Index. Plot 4 depicts the choice of 3 clusters, suggesting that the schools can be separated into: high, medium and low performance.

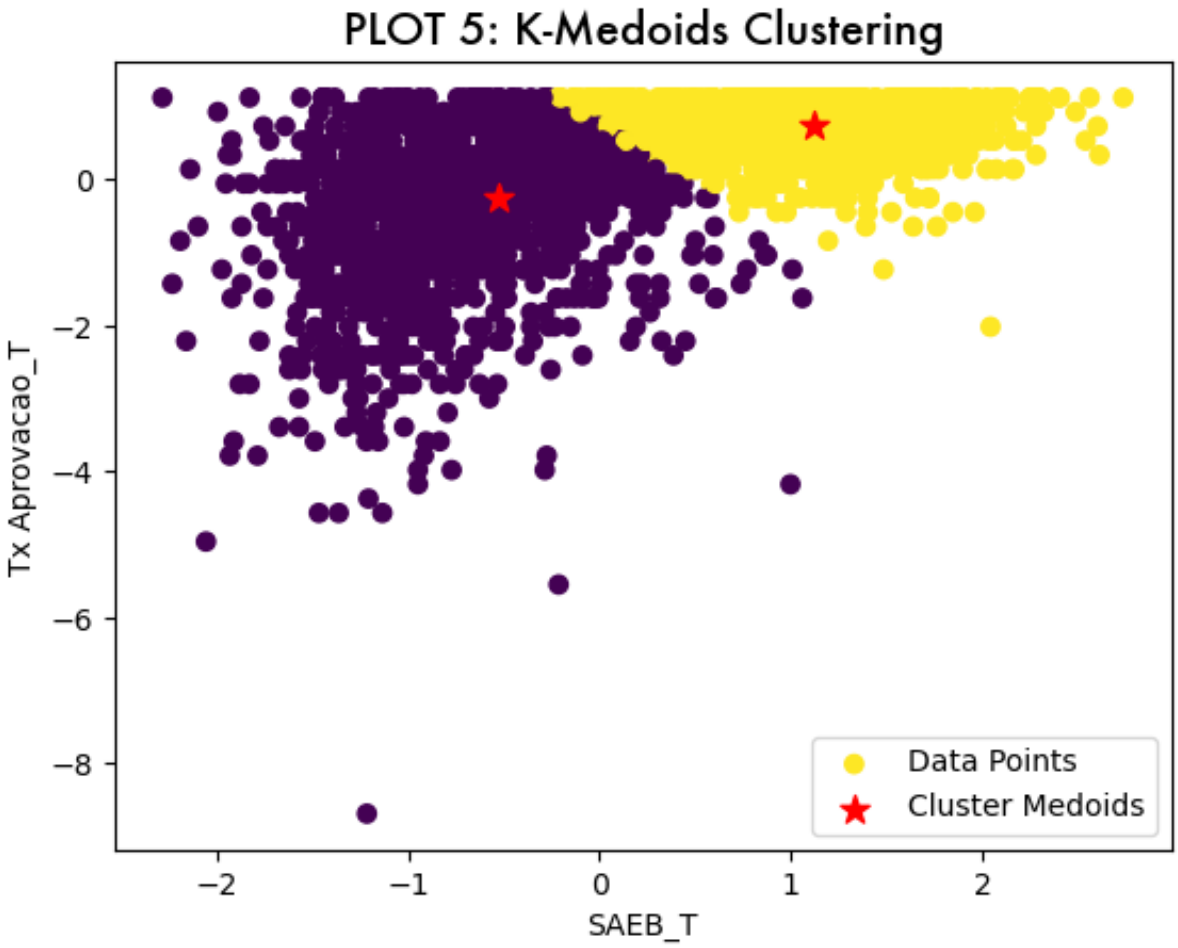
TABLE 1 - K-Means Scores

Number of Clusters	Silhouette Score	Davies-Bouldin Index
2	0.442286950155698	0.816091499379918
3	0.472179624020085	0.7549475972310069
4	0.432242252358546	0.8300654093086033

Besides dealing better with outliers, the most important difference between K-Medoids and K-Means is how they partition the dataset. Instead of using the mean of all data points within a cluster (centroids), K-Medoids uses the actual data points (medoids), that is how the results are less influenced by outliers (H, 2024). When K-Medoids is applied and tested (Table 2), the best option is the algorithm with 2 clusters (Plot 5) which is similar to Plot 2 with the raw data.

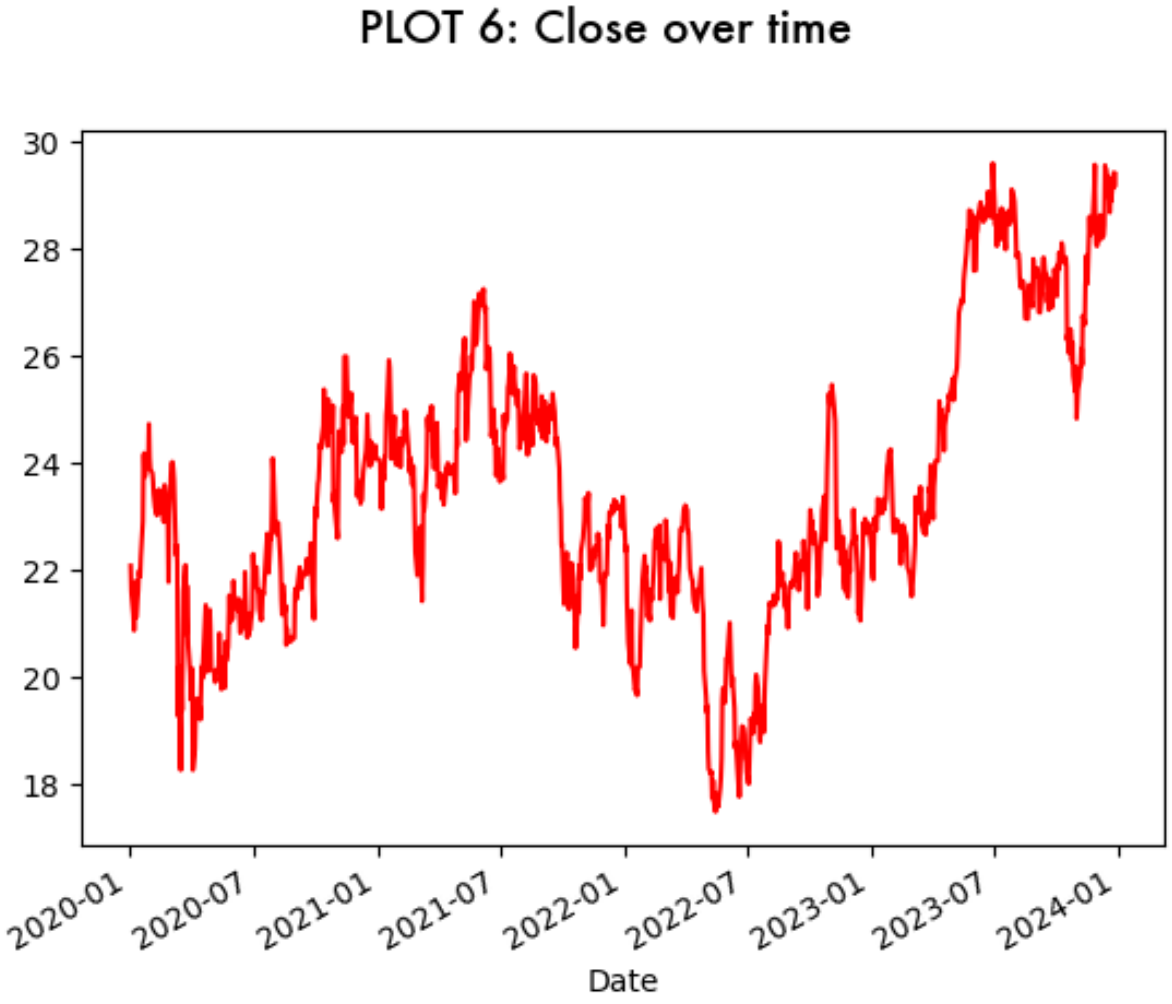
TABLE 2: K-Medoids Scores

Number of Clusters	Silhouette Score	Davies-Bouldin Index
2	0.45929437589950434	0.7538285487793357
3	0.44983446784705194	0.8090572600304085
4	0.39444352551840334	0.855939439014425



Times Series: ARIMA

The chosen period is from January 2020 to January 2024, which includes the COVID-19 pandemic. The feature to be analysed is the value of the stock when the market closes (Close) which is generally used to predict the future price. Plot 6 depicts the trend of the series, there are two major lows (around July 2020 and July 2022) and a considerable recovery after the second period. This plot is the first evidence of the non-stationarity of the time series.



A stationary time series assumes that the principal statistical measures are constant. The predictions are easier and better when mean, variance and autocorrelation do not change over time, making it possible to get the underlying dynamics (Robert Nau, Fuqua School of Business, Duke University, 2014).

The Dickey-Fuller (DF) test was developed to test if the mean and the variance of a time series are constant. An Augmented (ADF) version is often used in the financial field, where potential trends and serial correlations are accounted for. The null-hypothesis for both cases is the presence of a unit root meaning the non-stationary (Guo, 2023). The results of the ADF support the previous indication of non-stationary with the non rejection of the null hypothesis with a p-value of 0.2462 which is greater than the significant level of 0.05.

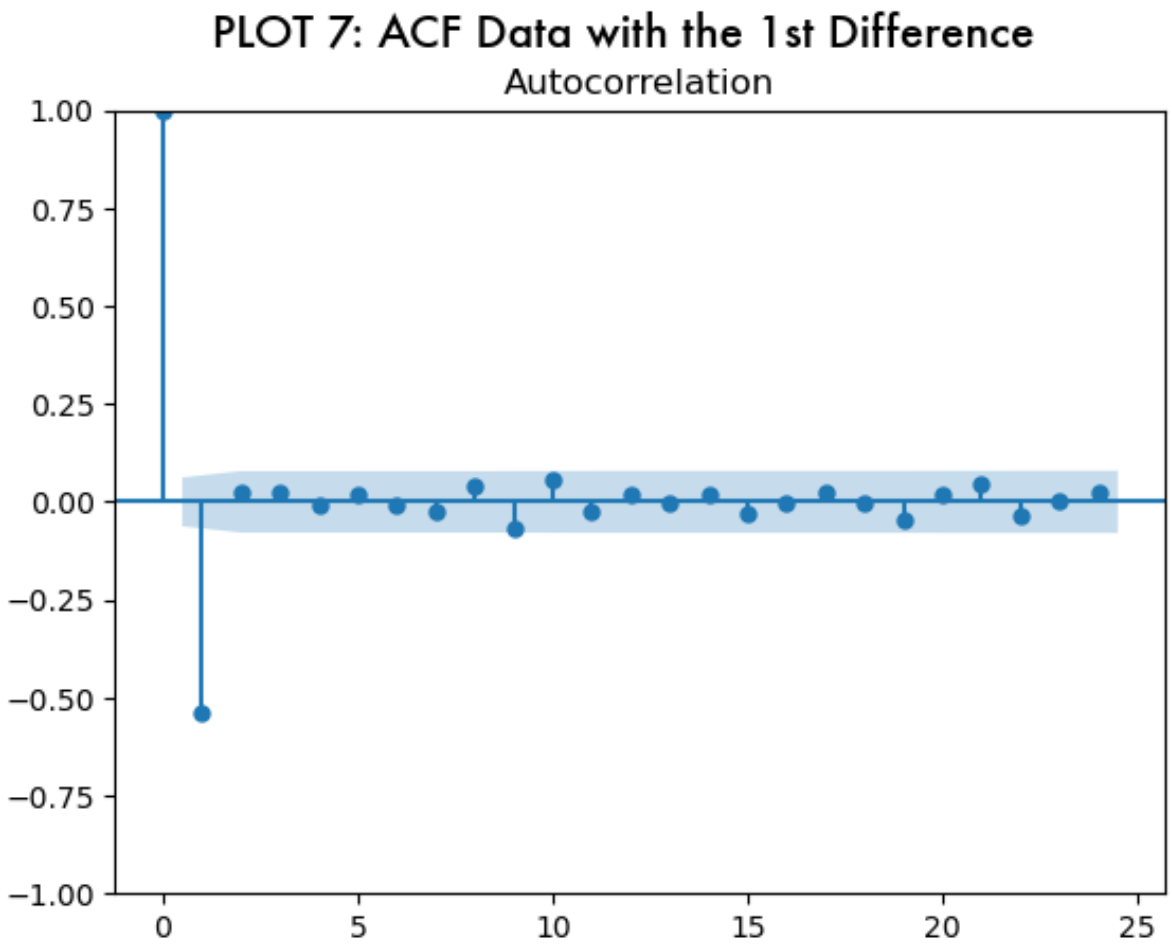
The First Difference is a mathematical method that helps to adjust the seasonality and to detrend the time series by making it more stationary. The calculations are based on the difference between consecutive observations in the series (Robert Nau, Fuqua School of Business, Duke University, 2014). The data became stationary after the first difference with a p-value of 0.0.

The ARIMA model is based on three parameters (Robert Nau, Fuqua School of Business, Duke University, 2014):

- AR (p): Autoregression is the assumption that future values can be predicted by the linear combination of past observations;
- I (d): Order of differentiation, or the number of times the time series has to be differentiated to become stationary;
- MA (q): Moving Average, the next observation is modelled based on the linear combination of the residual erros.

Plots 7 shows the autocorrelation of the time series with the first order of differentiation. The first two lags have the biggest spikes suggesting that the values for the parameters might be in the range of 0 and 2.

- p = 1 or 2, as the spikes get smaller after the second lag;
- d = 2, once the two spikes suggests that only one differentiation might not be enough;
- q = 0 or 1, as spikes at various lags does not necessarily means that the errors have an impact on future values, as the data itself has.



Three sets os values (p, d, q) were tested: (1, 2, 0), (1, 2, 1) and (2, 2, 1). Based on the visual plot and the statistical tests of the model, the second set (1, 2, 1) is the best one with the better fit and the p-values for the coefficients lower than the significant level of 0.05 as shown in Plot 8 and Table 3.

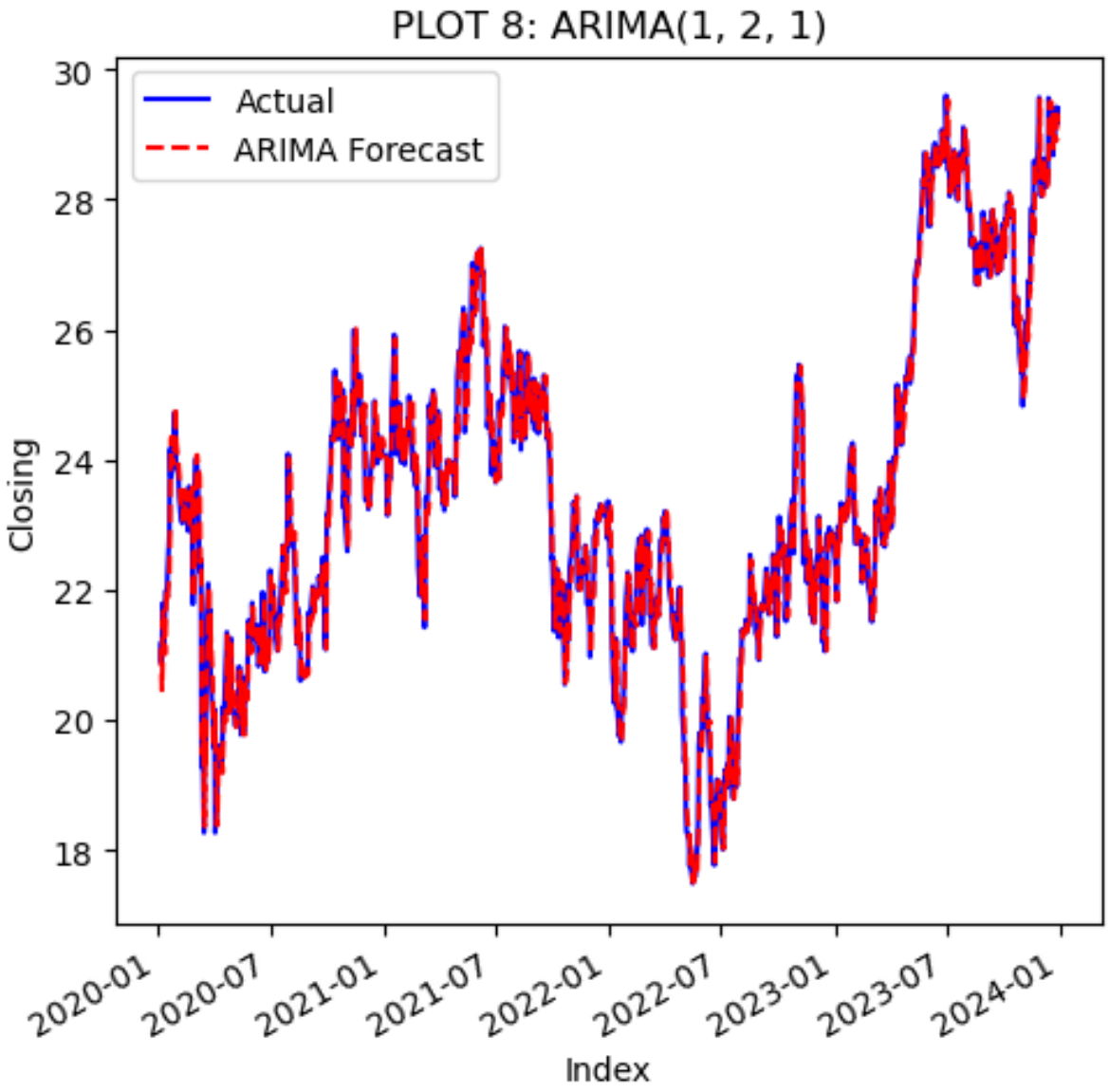


TABLE 3: ARIMA Statistical Summary

Dep. Variable:	y	**No. Observations:**	993
Model:	ARIMA(1, 2, 1)	**Log Likelihood**	-643.773
Date:	Mon, 22 Apr 2024	**AIC**	1293.547
Time:	18:21:50	**BIC**	1308.243
Sample:	0	**HQIC**	1299.135
	- 993		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	-0.0915	0.027	-3.369	0.001	-0.145	-0.038
ma.L1	-1.0000	0.413	-2.422	0.015	-1.809	-0.191
sigma2	0.2131	0.088	2.424	0.015	0.041	0.385

Ljung-Box (L1) (Q):	0.00	**Jarque-Bera (JB):**	253.55
Prob(Q):	0.97	**Prob(JB):**	0.00
Heteroskedasticity (H):	0.55	**Skew:**	-0.35
Prob(H) (two-sided):	0.00	**Kurtosis:**	5.38

Conclusion

Other methods for Clustering were tested such as Hierarchical Agglomerative, DBSCAN, Optics and Fuzzy C-Means. The results are provided in the main Jupyter notebook. The algorithms with the best results were K-Means and K-Medoids as explained earlier. Plots 4 and 5 confirmed the first assumption made with the raw data in Plot 2 that the schools are roughly grouped by its type, with a higher concentration of city schools with better approval and quality rates over the states school.

In Times Series analysis the stationary was reached and a stable model with the parameters (1, 2, 1) for Autoregression, Order of differentiation and Moving Average. There is space for a more complex discussion in future.

References

Aman (2021). *Hierarchical clustering Use Case In python | Hierarchical clustering example in Python*. [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=lQt92mh0N8I [Accessed 23 Apr. 2024].

Banerji, A. (2021). *K-Mean | K Means Clustering | Methods To Find The Best Value Of K*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/05/k-mean-getting-the-optimal-number-of-clusters/.

Governo Brasileiro (2023). *Saeb*. [online] Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira. Available at: https://www.gov.br/inep/pt-br/areas-de-atuacao/avaliacao-e-exames-educacionais/saeb.

Guo, Z. (2023). Research on the Augmented Dickey-Fuller Test for Predicting Stock Prices and Returns. *Advances in Economics Management and Political Sciences*, 44(1), pp.101–106. doi:https://doi.org/10.54254/2754-1169/44/20232198.

H, P.N. (2024). *Exploring the World of Clustering: K-Means vs. K-Medoids*. [online] Medium. Available at: https://medium.com/@prasanNH/exploring-the-world-of-clustering-k-means-vs-k-medoids-f648ea738508 [Accessed 20 Apr. 2024].

Hogg, G. (2022). *DBSCAN Clustering Coding Tutorial in Python & Scikit-Learn*. [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=VO_uzCU_nKw.

Infomoney (2024). *RADL3 (RAIADROGASIL ON NM)*. [online] InfoMoney. Available at: https://www.infomoney.com.br/cotacoes/b3/acao/rd-radl3/.

McDonald, A. (2021). *K-Means Clustering Algorithm with Python Tutorial*. [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=iNlZ3IU5Ffw.

Mulla, R. (2022). *Time Series Forecasting with XGBoost - Use python and machine learning to predict energy consumption*. *YouTube*. Available at: https://www.youtube.com/watch?v=vV12dGe_Fho.

Piech, C. (2013). *CS221*. [online] Stanford.edu. Available at: https://stanford.edu/~cpiech/cs221/handouts/kmeans.html.

Robert Nau, Fuqua School of Business, Duke University (2014). *Class 1:* [online] Available at: https://people.duke.edu/~rnau/Principles_and_risks_of_forecasting--Robert_Nau.pdf.

Ros, F., Riad, R. and Guillaume, S. (2023). PDBI: A partitioning Davies-Bouldin index for clustering evaluation. *Neurocomputing*, 528(0925-2312), pp.178–199. doi:https://doi.org/10.1016/j.neucom.2023.01.043.

Sharma, P. (2019). *The Most Comprehensive Guide to K-Means Clustering You'll Ever Need*. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/.

Six Sigma Pro SMART (2023). *K-Medoids Clustering | Cluster Analysis | Unsupervised Machine Learning | Data Science*. [online] www.youtube.com. Available at: https://www.youtube.com/watch?v=8PSbi6GKXrs [Accessed 23 Apr. 2024].

Tomar, A. (2023). *Stop Using Elbow Method in K-means Clustering | Built In*. [online] builtin.com. Available at: https://builtin.com/data-science/elbow-method#.

Appendix

Cluster Jupyter Notebook: http://localhost:8888/notebooks/Documents/GitHub/ML/SAEB.ipynb

Time Series Jupyter Notebook: http://localhost:8888/notebooks/Documents/GitHub/ML/RAIA.ipynb

XGB Jupyter Notebook: http://localhost:8888/notebooks/Documents/GitHub/ML/XGB.ipynb

Library References:

pandas McNuulty, J. (2010). Pandas: Powerful Python data analysis toolkit. John Wiley & Sons.: https://pandas.pydata.org/docs/

numpy Harris, C. R., Millman, K. J., van der Walt, S., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. Nature methods, 17(8), 849-854.: https://numpy.org/doc/

matplotlib Hunter, J. D. (2007). Matplotlib: A comprehensive system for creating scientific plots in Python. Computing in Science & Engineering, 9(3), 90-95.: https://matplotlib.org/

seaborn Waskom, M. L., & Hofmann, C. (2016). Seaborn: Factual statistical data visualization in Python. Journal of Open Source Software, 1(6), 26.: https://seaborn.pydata.org/

scikit-learn Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.: https://scikit-learn.org/stable/

sklearn-extra Wei, C.-Z., & Zhao, M. (2015). scikit-learn extensions: Going beyond the basics. Journal of Machine Learning Research, 16(1), 2225-2232.: https://scikit-learn-extra.readthedocs.io/en/stable/

scipy.cluster.hierarchy Agarwal, S., Guo, X., & Bar-Hillel, A. (2015). k-means clustering on spectral data. In Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS) (Vol. 1, pp. 456-464).: https://arxiv.org/abs/1109.2378

AgglomerativeClustering (from scikit-learn): Refer to scikit-learn documentation (https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html)

DBSCAN (from scikit-learn): Refer to scikit-learn documentation (https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.

KNNImputer (from scikit-learn): Refer to the scikit-learn documentation for `KNNImputer` (https://scikit-learn.org/stable/modules/generated/sklearn.impute.KNNImputer.html)

Normalizer (from scikit-learn): Refer to the scikit-learn documentation for `Normalizer` (https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.Normalizer.html)

silhouette_score (from scikit-learn): Refer to the scikit-learn documentation for `silhouette_score` (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html)

plotly.express (plotly): Cook, P., Gandrud, C., Fokkema, M., & Peterson, P. (2017). Plotly for Python: A Primer. Packt Publishing Ltd.: https://plotly.com/python/

itertools (built-in Python module): No formal reference needed as it's a core Python library.

gridspec (from matplotlib): Hunter, J. D. (2007). Matplotlib: A comprehensive system for creating scientific plots in Python. Computing in Science & Engineering, 9(3), 90-95.: https://matplotlib.org/

make_blobs (from scikit-learn): Refer to the scikit-learn documentation for `make_blobs` (https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_blobs.html)

scatterplot (from seaborn): Refer to the seaborn documentation for `scatterplot` (https://seaborn.pydata.org/generated/seaborn.scatterplot.html)

plot_acf (from statsmodels): Seabold, P., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with Python. Proceedings of the 9th Python in Science Conference (Vol. 57, pp. 61-67).: https://www.statsmodels.org/stable/

plot_pacf (from statsmodels): Seabold, P., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with Python. Proceedings of the 9th Python in Science Conference (Vol. 57, pp. 61-67).: https://www.statsmodels.org/stable/

ARIMA (from statsmodels): Hyndman, R. J., & Athanasopoulos, G. (2013). Forecasting: principles and practice (Vol. 67). OTexts.: https://www.statsmodels.org/stable/

statsmodels.api (from statsmodels): Seabold, P., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with Python. Proceedings of the 9th Python in Science Conference (Vol. 57, pp. 61-67).: https://www.statsmodels.org/stable/

statsmodels.tsa.api (from statsmodels): Seabold, P., & Perktold, J. (2010). Statsmodels: Econometric and statistical modeling with Python. Proceedings of the 9th Python in Science Conference (Vol. 57, pp. 61-67).: [https://www.statsmodels.org/stable/](https