**CCT College Dublin**

**Assessment Cover Page**

*To be provided separately as a word doc for students to include with every submission*

|  |  |
| --- | --- |
| **Module Title:** | MSc in Data Analytics |
| **Assessment Title:** | Integrated CA2 Sem 2 MSc in Data Analytics  A Time-series Forecast of Tweet Sentiment Scores |
| **Lecturer Name:** | David McQuaid  Muhammad Iqbal |
| **Student Full Name:** | Deniz Ozsoy Ugur |
| **Student Number:** | 2023195 |
| **Assessment Due Date:** | 19/05/2024 |
| **Date of Submission:** | 19/05/2024 |
| **Word Count** | 3343 (excluding references, titles, citations and quotes) |
| **GitHub Repository** | CCT-Dublin Link:  <https://github.com/CCT-Dublin/adv-data-big-data-ft-ca2-denizozsoyugur>  The link used since the beginning of the project:  <https://github.com/denizozsoyugur/2023195_IntegratedCA2> |

**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. |

# Abstract

*This article examines a dataset of tweets on Twitter. These are stored in a MySQL database and sentiment scores are generated using Apache Spark and the data is then stored using Apache Hive. Sentiment scores are then used in a time series analysis and predictions are made over short periods of 1 day, 3 days and 7 days. Two time series models were used, ARIMA and LSTM; ARIMA was chosen as the best performing method and was used as the basis for an interactive dashboard created using Dash in Python.*

**Keywords:** MySQL, Apache Spark, Apache Hive, Big Data Processing, Sentiment Analysis, Time Series Analysis, ARIMA, LSTM

# Introduction

Twitter, now called X, is a social media platform where users can post messages of 140 characters or less, among other types of media. Tweets can be about any topic and can be viewed or reshared by other users. A dataset of 1.6 million tweets was provided, including tweet, user, and tweet date. Sentiment Analysis is a subfield of Natural Language Processing. This methodology has attracted the attention of many researchers because it allows the investigation of various problems in social media. It is aimed to create sentiment analysis of tweets with this information. This will be accomplished as the data is processed and moved from one database to another using big data storage and processing techniques such as Apache Spark and Hadoop Distributed File System. The sentiment score will then be analysed and used as the basis of time series analysis. This time series analysis will look at two different models, ARIMA and LSTM, where parameter tuning, and predictions are made with both. Since a short-term time series will be created, the LSTM model was preferred to obtain sensitive results for sensitivity score and pollution reduction. The most effective model will be selected to predict the sentiment of future tweets with 1-day, 3-day and 7-day time intervals. These future predictions will then be plotted and displayed using an interactive control panel, allowing selection of the time period containing the predicted values.

# Data Analysis Process

In Data Analysis process, a plan was made based on the CRISP-DM model. The Cross Industry Standard Process for Data Mining (CRISP-DM) represents the most common basic methodology used to standardise data mining processes in all sectors (Hotz, 2018) (Fig.1)

1. Business/Research Understanding Phase

2. Data Understanding Phase

3. Data preparation Phase

4. Modelling Phase

5. Evaluation Phase

6. Deployment Phase

A diagram of a business process

Description automatically generated

Fig.1 CRISP-DM

Understanding the goals of the project, defining the business problem, and determining success criteria. Relevant data sources are identified, data quality is evaluated, and initial insights are obtained. Cleaning, transforming, and preparing data for modelling. The performance of the model is evaluated according to the success criteria defined in the Business Concept phase. The model is deployed in a production environment and its performance is monitored. Each phase builds on the previous one and is critical to the success of the project. The Business Understanding phase is critical to the success of the CRISP-DM methodology. Ensures that the project is aligned with business objectives, identifies relevant data sources, and establishes success criteria. A comprehensive Business Understanding phase can benefit the project, including improved outcomes, reduced risk, efficient use of resources, and improved communication.

## Data Understanding / Initial Loading

The data source for this project is a csv file called 'ProjectTweets.csv' provided by CCT College's moodle page at 'https://moodle.cct.ie/mod/assign/view.php?id=44089'. When loaded as a data frame into a Python environment using the Pandas library, it contains 1.6 million tweets extracted from April to May 2009, with columns for the index, date, time, user, the tweet text itself, and others.

Since the data source will first be stored in a MySQL database before processing, an initial processing is performed in which special characters are removed from the text column; This is to avoid any allocation issues while loading data. The modified csv file is saved as 'ProjectTweets2'.

A graph showing the number of tweets

Description automatically generated

Fig 2, Hourly Tweet Count from EDA

This csv file is then loaded into the MySQL database called 'Tweets', into a table called 'usertable' created for this purpose. The table is created with five VARCHAR columns to store the dataset. It is stored directly via the MySQL terminal using a 'DATA INFILE' command to store the entire CSV at once, ignoring the first line as it contains the header names.

"#sudo mv /home/hduser/2023195\_IntegratedCA2/ProjectTweets\_2.csv /var/lib/mysql-files/

#LOAD DATA FILE '/var/lib/mysql-files/ProjectTweets\_2.csv' TO TABLE usertable ',' FIELDS TERMINATED WITH ',' OPTIONALLY ENDED WITH '"' ROWS '\n' CONSIDER 1 LINE; "

Retrieving the tweet dataset from MySQL database using Apache Spark was successful. Implementation of UDFs, including the sentiment analysis model, was also successful, and checking the "sentiment\_score" data type showed that it was a string, converted to a floating-point type. The new dataframe is written to the Apache Hive database. Depending on the library selection I used, the sentimented score appeared in the range of 0-100. Plotting the sentiment scores using a boxplot shows that the average sentiment is above 50, meaning the trend is positive, but there are a greater number of extreme outliers on the negative side of the score. Values below 50 are perceived as negative, 50 as neutral and above 50 as positive.(Fig 2)

A graph with lines and numbers

Description automatically generated

Fig 3, Boxplot of General Sentimented score

Sentiment score metrics are records as follows:

|  |
| --- |
| Mean: 56.815873 |
| Min: 55.279662 |
| 25%: 56.276359 |
| 50%: 56.899812 |
| 75%: 57.369476 |
| Max: 57.977270 |

A graph showing a graph

Description automatically generated with medium confidence

Fig 4, Line plot of sentiment score over time

The overall sentiment score was also plotted.

A graph showing the value of a certain time

Description automatically generated with medium confidence

Fig 5, Sentiment Score Tweet Plot

## Database Comparison

For this project, open-source software Yahoo! Cloud Serving Benchmark (YCSB) offers us the opportunity to test the performance of different databases that perform read and write operations. The YCBS system tracks various metrics and generates a report. Yahoo! The Cloud Service Benchmark (YCSB) tool allows comparison of database types based on a variety of different criteria using custom or provided sample datasets. In this case, testing will be done on MySQL database against MongoDB NoSQL database. The 'workload' provided in this case is used because it is "a 50/50 mix of reading and writing" (Busbey, 2020), which suits the type of work we will be using.

Both MySQL and MongoDB services are started via command terminal and a MySQL table is created to store the sample workload, MongoDB will create the table during installation.

Once the YCSB service is configured, the workload is executed against two different databases and the results are stored in .txt files for analysis. The testing was done on a virtual machine with Ubuntu 22.04 installed. MongoDB and MySQL were then installed on this virtual machine.

## Sentiment Analysis

Data is retrieved from MySQL database using Apache Spark running in Jupyter notebook. Spark was chosen because it is "an open-source data processing engine for large datasets" (IBM, n.d.) that will allow loading, processing, and storing of Twitter data.

A Spark session is created, MySQL connection settings are created, and the dataset is loaded from 'usertable' into the Spark dataframe.

A sentiment analysis function is created using the 'HuggingFace' library to import a sentiment analysis pipeline using the 'Distilbert-base-uncased-finetuned-sst-2-english' model.

Pipelines consist of a tokenizer, which is “responsible for matching the raw text input to the token” (Hugging Face, 2023); This is a model that then makes predictions about tokenized inputs and optional post-processing arguments. The function containing this line is then saved as a user-defined function (UDF) in the Spark session. A second UDF is created and saved that will truncate the date column to include only the day, month, and year and remove the time zone. (Mon Apr 06 2009). Some of the relevant operations were performed in Ubuntu and some in Windows interface.

After creating the Spark df containing the tweet dataset, two sentiment and date processing UDFs are applied to the 'text' and 'date' columns respectively. The resulting columns are "sentiment\_score" and "full\_date". The 'sentiment\_score' column was changed from string to float using the 'cast' function.

Taking the created dataset, an initial overview of it shows 1,598,314 rows with 2 columns. The ‘full\_date’ column is converted from a string to datetime, and a list of missing dates is created based on the first and last date contained in the dataset. The missing dates are added, with the corresponding sentiment scores being set to NA. The data is then grouped by the date column, aggregating the ‘sentiment\_score’ column by mean value.

This aggregated set is then checked for NA sentiment scores, and those that are missing are filled using linear interpolation, “used to estimate unknown values that lie between known values, [particularly for] evenly spaced data”.

## Time Series Analysis

Before creating the time series analysis, some tests were carried out to check whether the data were stationary or not, as well as trend and seasonality. “A stationary time series has no mean, variance, autocorrelation, etc. It is a series whose statistical properties such as are all constant over time” (McKinney W. 2012).

Dickey-Fuller test was applied to the data and the results showed that the data was stationary. To be sure, KPSS test was performed, and it was examined that the data was stationary.

Seasonality, trends, and residuals are plotted and a seasonally disaggregated column containing sentiment scores is created. The generated columns are tested to find which one has the lowest p-value and is therefore the most stationary.

Results of Dickey-Fuller Test:

Test Statistic -2.891555

p-value 0.046339

#Lags Used 5.000000

Number of Observations Used 75.000000

Critical Value (1%) -3.520713

Critical Value (5%) -2.900925

Critical Value (10%) -2.587781

dtype: float64

The time series is stationary.

A graph showing the value of a product

Description automatically generated with medium confidence

Fig 6, Line plot of stationary sentiment score, rolling mean and rolling average over time

A graph showing a line of a person

Description automatically generated with medium confidence

Fig 7, Autocorrelation

A graph of a function

Description automatically generated

Fig 8, Autocorrelation and Partial Autocorrelation

According to the KPSS test results, it can be said that the time series is stationary and does not contain any statistically significant change. This means that the tie series remains constant around a certain mean and variance over time.

KPSS Test:

Test Statistic: 0.337701523120429

p-value: 0.1

Lags Used: 4

Number of Observations: {'10%': 0.347, '5%': 0.463, '2.5%': 0.574, '1%': 0.739}

Result: The time series is stationary as the p-value is greater than the significance level (0.05).

After seasonal decomposition, the p-value became 2.787851e-07 which was even more stationary and the reason this set of values was chosen for the model.

A graph showing a graph

Description automatically generated with medium confidence

Fig 9, Line plot of seasonally decomposed sentiment scores, rolling mean and rolling average

over time

## ARIMA

The Autoregressive Integrated Moving Average (ARIMA) model is a popular time series forecasting model that can represent several different types of time series and is particularly suitable for “data differentiated to achieve weak stationarity” (Lim, 2019)

To try to improve the model's results, the data is normalized using the MinMaxScaler() function. The data was then split into training and testing sets, with 65 days in the training set and 16 days in testing; It was roughly an 80/20 split.

There are a few different approaches and strategies for hyperparameter tuning, with no established standard specifically for time series analysis. Some approaches include a grid search, “a brute force approach constrained by a set of predefined combinations of hyperparameters, i.e., grid points” (Bakhashwain and Sagheer, 2021), and random search, where predefined sets of hyperparameters are typically randomly and evenly sampled. It leads to better learned models with less computational time.

To find the best parameters for the ARIMA model, a function is created by testing a range of p, d, and q values and returning the combination that provides the best result. This is run on the training set and the results are applied as parameters in the ARIMA model.

The model is then built and adapted using the training data. A report is printed, and the residuals are plotted. Predictions are made based on the test set; the result is plotted, and the Root Mean Square Error (RMSE) is calculated.

Predictions are also plotted against the entire dataset to visualize the variance. Using the created model, predictions were made for the future in 1-day, 3-day and 7-day periods, respectively. These were then visualized as an extension of the existing time series data.

### Hyperparameter Results and Model

The p, d, q value combination found to give the best model was ‘(2, 0, 1)’. These were applied to the ARIMA model used. When tested on the test set, the model produced a RMSE of 0.141686 and an R2 score of -6.651820 A low RMSE error is good, indicating a good prediction of values, however the negative R2 score indicates that the model is not fitted correctly, does not explain the variance in the original data, and therefore is not fit for predictions.

A low RMSE error is good, indicating a good prediction of values, however the negative R2 score indicates that the model is not fitted correctly, does not explain the variance in the original data, and therefore is fit for predictions.

A graph showing a graph

Description automatically generated with medium confidence

Fig 10, Best Arima Model Actual and Predicted

Forecasts are created and drawn for 1-day, 3-day and 7-day periods. Negative R2 score issues become apparent when looking at the forecasts as the forecasts remain essentially constant at 0.000 for the entire time frame.

## LSTM

LSTM is a type of artificial neural network designed to identify patterns in sequential data over time, such as time series. It can capture long-term dependencies through memory cells, making it particularly powerful in time series analysis. (P.Lee,2022) This model learns patterns in historical data and tries to predict future values based on these patterns. Tweet dataset was chosen for short-term forecasting, especially because this method is sensitive, and it has been researched that it will increase the forecast accuracy. Emotional analysis from tweet series includes sensitivity according to the machine's learning status.

A chart with green and blue squares

Description automatically generated

Fig 11, LSTM Prediction Model

In the LSTM model, Epochs are used to organize the training process of the model. Each epoch means that all samples in the training dataset are seen once by the model. This allows the model to see the entire dataset and learn general patterns in the dataset.

By determining the number of periods, it was determined how long the model would continue the education process and the model was established. Periods are also used to evaluate the performance of the model. During training, at the end of each epoch the model can be tested on a separate "validation" data set. This validation dataset consists of data that the model has not seen before and is often used to measure the generalization ability of the model. Inter-period validation results were examined to check whether there were problems such as overfitting or underfitting in the model. In LSTM and other deep learning models, periods are the basic building blocks of the educational process. They are important for the model to learn patterns in the dataset, evaluate its performance, and optimize the training process. It can prevent undesirable situations (such as overfitting) by increasing the generalization ability of the model. Therefore, the LSTM model was chosen for this prediction.

A graph showing a number of different types of performance

Description automatically generated with medium confidence

Fig 12, LSTM Model Performance

Since careful selection and monitoring of the periods in the training of the model will be critical for the success of the model, looking at the number of observations and periods in the last year of this model, we can say that the past data are compatible with the model. However, at the same time, it can be said that ARIMA is the most successful model of the future in the prediction of future periods and of the two models used.

## DASHBORD

A dashboard can be ideal for viewing multiple items or filtering between different slices of the same item. Ideally, “[the dashboard] will support analytical and communicative goals such as monitoring and reporting through functions such as interaction and storytelling” (Setlur et al., 2023)

A graph with a line

Description automatically generated

Fig 13, LSTM Dashbord

## Database Comparison

A comparison of MySQL and MongoDB was made for the initial selection of the database that would store 'ProjectTweets.csv'. MongoDB is a NoSQL (not just SQL) database. It is used to access and analyse large amounts of unstructured data and can do this because it does not store information in a traditional relational format, but instead processes individual items. MySQL requires a predefined schema for each table, which can be more rigid and less flexible, but in this case the data is simple and flat and fits well with the MySQL structure.

The results of the database comparison indicate that for the MySQL database;

Load Time: 7.2032985367191162 seconds

Read Time: 29.8856569380805969 seconds

Write Time: 9.81154554514016724 seconds

And for the MongoDB database;

Time to write to MongoDB: 52.788695665561 seconds

Time to read from MongoDB: 41.8554866241577 seconds

When using PySpark, both databases can be accessed with read and write capabilities. MySQL is generally faster at selecting data from a table than MongoDB is from a collection, especially if the data is indexed and the queries are simple, in which case both are true. For these reasons, MySQL was chosen as the database to store the initial tweet dataset.

Apache Spark was chosen as the data processing tool because it could be easily integrated with Python using PySpark. PySpark, the Python API for Apache spark, enables the performance of “real-time, large-scale data processing in a distributed environment using Python” (Apache, 2023).

Apache Hive was also chosen as a data warehouse tool. Hive is built on top of Hadoop and supports storage via HDFs. It allows users to “read, write, and manage petabytes of data using SQL” (Apache Software Foundation, 2013).

There may be a variety of factors to consider when choosing between machine learning models. The interpretability of a model is how explainable the result is. ARIMA models provide a clear explanation of how past values and error terms affect the current value. (Simon, Glaum and Valdovinos, 2023).

The models have a wider selection of hyperparameters that can be tuned to improve the performance of the model. While the number of trees, maximum depth, minimum node size and delay are some of the parameters, there are only autoregressive, diverging and moving average components in ARIMA models.

1-Day Predictions:

[0.54948196]

3-Day Predictions:

[0.5494819648570468, 0.35145705619030265]

7-Day Predictions:

[0.5494819648570468, 0.35145705619030265, 0.4589747411846078, 0.5693929532226576, 0.35256811725012194, 0.4592469036641639]

## CONCLUSION

The initial dataset went through a series of transformations where it was stored entirely in a MySQL database before being processed and reduced using Apache Spark and a sentiment analysis model programmed from a Juypter notebook using the PySpark and Hugging Face 'pipeline' package. This data was then stored using Apache Hive and accessed again to perform a time series analysis on the extracted sentiment scores. When the scores were initially plotted over time, it became clear that there were significant differences in the sentiment of the tweets. In terms of Sentiment Analysis, the sentiment distribution appeared to be relatively balanced, with similar tweet density for negative, positive and neutral sentiments. Most tweets were classified as neutral, followed by approximately equal proportions of positive and negative tweets.

For Time Series Analysis, Seasonal decomposition of sentiment data did not reveal any significant seasonal patterns, indicating the absence of recurring sentiment patterns over time. Time series data exhibited a non-stationary structure due to the presence of a trend.

Autocorrelation analysis showed that there was no significant correlation between positive and negative emotions at different time lags. Daily average sentiment charts showed changing sentiment patterns over time, with peaks and troughs observed over different periods. Average positive emotion was skewed towards a positive emotion, while average negative emotion was skewed towards a negative emotion. Daily deviation sensitivity charts highlighted the high variability in sensitivity values over different periods.

|  |  |  |
| --- | --- | --- |
| PREDICTION | ARIMA | LSTM |
| 1 Day Prediction | 0.098604 | 0.549482 |
| 2 Day Prediction | 0.041678 | 0.351457 |
| 3 Day Prediction | -0.01542 | 0.355625 |
| 4 Day Prediction | -0.07803 | 0.458975 |
| 5 Day Prediction | -0.07445 | 0.569393 |
| 6 Day Prediction | -0.07723 | 0.352568 |
| 7 Day Prediction | -0.0445 | 0.459247 |

The LSTM model performed well in predicting future sentiment scores and was decided to be the model used to generate the predicted values to be used in the final dashboard.

It is possible that data are insufficient over time and the lack of explanatory variables makes future predictions difficult.

Analysis of sentiment and time series patterns in the tweet dataset provided information about the distribution of sentiment, sentiment trends over time, and the presence of variability. These findings may be useful for understanding the dynamics of public sentiment and for further research and forecasting in sentiment and time series analysis.

In terms of the benchmark tool and according to the load test results, MongoDB performed better in several aspects. MongoDB exhibited shorter uptime, higher throughput, lower average write latency, lower maximum write latency, lower 95th percentile latency, and no errors during testing. These results show that MongoDB performs better in terms of overall performance and stability under the given load testing conditions.

## REFERENCES

*[Duke University (2019). Stationarity and differencing of time series data. [online] Duke.edu. Available at: https://people.duke.edu/~rnau/411diff.htm [Accessed 5 Nov. 2023].*

*Bakhashwain, N. and Sagheer, A. (2021). Online Tuning of Hyperparameters in Deep LSTM for Time Series Applications. International Journal of Intelligent Engineering and Systems, 14(1), pp.212–220. doi:https://doi.org/10.22266/ijies2021.0228.21.*

*Győrödi, C.A., Dumşe-Burescu, D.V., Zmaranda, D.R. and Győrödi, R.Ş. (2022). A Comparative Study of MongoDB and Document-Based MySQL for Big Data Application Data Management. Big Data and Cognitive Computing, 6(2), p.49. doi:https://doi.org/10.3390/bdcc6020049.*

*[Rashid, A. (2013). Learning MongoDB. Packt Publishing.*

*[Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and Practice (2nd ed.). OTexts. Retrieved from https://otexts.com/fpp2/*

*Internet Archive: About IA (no date). Available at: https://archive.org/about/ (Accessed: 29 April 2023).*

*Sanh, V. et al. (2020) ‘DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter’. arXiv. Available at: http://arxiv.org/abs/1910.01108 (Accessed: 18 May 2023).*

*Apache (2023). PySpark 2.4.5 documentation. [online] spark.apache.org. Available at: https://spark.apache.org/docs/latest/api/python/index.html [Accessed 6 Nov. 2023].*

*Apache Software Foundation (2013). Apache Hive TM. [online] Apache.org. Available at: https://hive.apache.org/ [Accessed 6 Nov. 2023].*

*Azis, A., Fariz Darari and Muhammad Rizqy Septyandy (2020). Time Series Analysis on Earthquakes Using EDA and Machine Learning. Azis, M. F. A., Darari, F., & Septyandy, M. R. (2020). Time Series Analysis on Earthquakes Using EDA and Machine Learning. 2020 International Conference on Advanced Computer Science and Information Systems (ICACSIS). doi:https://doi.org/10.1109/icacsis51025.2020.9263188.*

*Bakhashwain, N. and Sagheer, A. (2021). Online Tuning of Hyperparameters in Deep LSTM for Time Series Applications. International Journal of Intelligent Engineering and Systems, 14(1), pp.212–220. doi:https://doi.org/10.22266/ijies2021.0228.21.*

*Lim, Y.H. (2019). SAAS Berkeley. [online] saas.berkeley.edu. Available at: https://saas.berkeley.edu/rp/arima [Accessed 5 Nov. 2023].*

*Greg Schulz, 2012, Cloud and Virtual Storage Networking, CRC Press.*

*Big Data Systems: A 360-degree Approach, Jawad Ahmed Shamsi, Muhammad Ali Khojaye, CRC Press, 2021.*

*Spark in Action, Second Edition, Jean-Georges Perrin, Rob Thomas, Manning Publishers, 2020*

*https://hadoop.apache.org/docs/stable/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html*

*Busbey, S. (2020). Core Workloads · brianfrankcooper/YCSB Wiki. [online] GitHub. Available at: https://github.com/brianfrankcooper/YCSB/wiki/Core-Workloads [Accessed 5 Nov. 2023].*

*Columbia University (n.d.). Interpolation Techniques. [online] iridl.ldeo.columbia.edu. Available at: https://iridl.ldeo.columbia.edu/dochelp/StatTutorial/Interpolation/index.html#Linear [Accessed 5 Nov. 2023].*

*Duke University (2019). Stationarity and differencing of time series data. [online] Duke.edu. Available at: https://people.duke.edu/~rnau/411diff.htm [Accessed 5 Nov. 2023].*

*Győrödi, C.A., Dumşe-Burescu, D.V., Zmaranda, D.R. and Győrödi, R.Ş. (2022). A Comparative Study of MongoDB and Document-Based MySQL for Big Data Application Data Management. Big Data and Cognitive Computing, 6(2), p.49. doi:https://doi.org/10.3390/bdcc6020049.*

*Hugging Face (2023). Pipelines. [online] huggingface.co. Available at: https://huggingface.co/docs/transformers/main\_classes/pipelines [Accessed 7 Nov. 2023].*

*Hyndman, R.J. and Athanasopoulos, G. (2018). Forecasting : Principles and Practice. 2nd ed. Heathmont, Vic.: Otexts.*

*IBM (n.d.). What is Apache Spark? | IBM. [online] www.ibm.com. Available at: https://www.ibm.com/topics/apache-spark [Accessed 5 Nov. 2023].*

*Lim, Y.H. (2019). SAAS Berkeley. [online] saas.berkeley.edu. Available at: https://saas.berkeley.edu/rp/arima [Accessed 5 Nov. 2023].*

*Ortiz, J.A.R. and J.E. (2021). Skforecast Docs. [online] skforecast.org. Available at: https://skforecast.org/0.10.1/index.html [Accessed 5 Nov. 2023].*

*Plotly (2023). Dash Documentation & User Guide | Plotly. [online] dash.plotly.com. Available at: https://dash.plotly.com/.*

*Setlur, V., Correll, M., Satyanarayan, A. and Tory, M. (2023). Heuristics for Supporting Cooperative Dashboard Design. IEEE Transactions on Visualization & Computer Graphics (Proc. IEEE VIS). [online] Available at: http://vis.csail.mit.edu/pubs/cooperative-dashboards [Accessed 7 Nov. 2023].*

*Simon, S., Glaum, P. and Valdovinos, F.S. (2023). Interpreting random forest analysis of ecological models to move from prediction to explanation. Scientific Reports, 13(1). doi:https://doi.org/10.1038/s41598-023-30313-8.*

*Timeseries — darts documentation (no date). Available at: https://unit8co.github.io/darts/generated\_api/darts.timeseries.html (Accessed: 23 May 2023).*

*Verma, Y. (2021) Complete Guide To Dickey-Fuller Test In Time-Series Analysis, Analytics India Magazine. Available at: https://analyticsindiamag.com/complete-guide-to-dickey-fuller-test-in-time-series-analysis/ (Accessed: 24 May 2023).*

*Prasad, A., 2021. Importance of Machine Learning in Making Investment Decision in Stock Market. s.l.:s.n.*

*P. Lee, Z. H. Y. T., 2022. Trend Prediction Model of Asian Stock Market Volatility Dynamic Relationship Based on Machine Learning. s.l.:s.n.*

*McKinney W. 2012, Python for Data Analysis, O'Reilly Media [ISBN: 13: 978-14493]*