# COMP6237 - Predicting the Performance of the UK Luxury Fashion Industry (Group 9)

**Joel Edgar**
University of Southampton
Southampton, United Kingdom
je1g19@soton.ac.uk

**Benjamin Sanati**
University of Southampton
Southampton, United Kingdom
bes1g19@soton.ac.uk

**Adam Ali**
University of Southampton
Southampton, United Kingdom
aka1g19@soton.ac.uk

**Sayedur Khan**
University of Southampton
Southampton, United Kingdom
srk1g19@soton.ac.uk

**Adwaith Kakkadath Palat**
University of Southampton
Southampton, United Kingdom
akp1g19@soton.ac.uk

## ABSTRACT
The UK luxury fashion industry is a major economic sector that has the potential to make a substantial contribution to the macroeconomic state of the UK. This paper outlines the methodology used to forecast the performance of the UK luxury fashion industry by using traditional macroeconomic indicators with the extraction of sentiment towards the UK government from social media platforms. The data is subsequently preprocessed and used with a variety of time-series models to forecast the industry's performance. The results show that Prophet was most successful in predicting performance, achieving a Mean Absolute Percentage Error (MAPE) of 11.36%.

### Author Keywords
Datamining; prophet; time-series; XGBoost

### CCS Concepts
•**Computing methodologies** → Supervised learning;

## INTRODUCTION
Predicting the performance of various sectors of industry in the United Kingdom (UK) has a range of benefits. Whether you are a member of the government working to guide economic policy or a private investor performing due diligence before making a decision on purchasing shares in a company; having an idea of how various companies will perform in the future provides essential insight to any decision-making process.

In this report, the authors investigate how various economic indicators can be used to predict the performance of the UK luxury fashion sector. To begin, the features used in the models are revealed to the reader and the reasoning for their usage is explained. Subsequently, a section explaining the data preparation and feature engineering details the data preparation process for use in the chosen models, followed by a section on exploratory data analysis, showcasing how the features had their importance ranked to help choose the correct level of complexity for corresponding models. Finally, the process for training and tuning the hyperparameters of 4 different types of models is explained, followed by a comparison of the models in the conclusion.



*CHI'20,* April 25–30, 2020, Honolulu, HI, USA



ISBN 978-1-4503-6708-0/20/04...$15.00
DOI: https://doi.org/10.1145/3313831.XXXXXXX

## COLLECTING DATA
The following economic indicators are used as features for the models in this report (all of these are for the UK only): GDP, unemployment, productivity, Consumer Price Index (CPI), CPIH, interest rate [2] and core inflation. The idea is by using economic indicators we can predict the health of the UK and in turn predict the willingness of customers to spend on frivolous luxury goods. A final feature called "Sentiment" was also included in the dataset. This feature is used to capture public opinion toward the government. All of the data sets were collected in the time frame from March 2013 to March 2023.

The authors decided to use a monthly sampling frequency for the analysis of the UK luxury fashion industry. However, not all of the data sources contained data that obeyed by this sampling frequency. To address this issue, the pandas interpolate method was used to interpolate between (a maximum of quarterly) data. This process produced a dataset with a constant sampling frequency throughout a specified time- frame, ranging from March 2013 to March 2023.

### Sentiment
Many of the features, e.g. core inflation and CPI/H are themselves produced through models rather than directly observed data. The authors of the paper thought it would be beneficial to include a feature that captures the attitude of customers as close to the source as possible. This discussion led to the

creation of the sentiment feature which, by analysing the sentiment toward the UK government, can be used to identify periods of unrest or low confidence in the political state of the UK. In doing so, the authors hypothesise that the impact of political factors on the population influences consumer behaviour, specifically their likelihood to spend money on luxury items.

The dataset for the sentiment was collected by the authors of the paper rather than from a third party because a suitable one could not be found online. Data collection started with Twint [17], a free scraping tool that uses the Twitter API and the advanced search function to gather tweets on a specific topic. By refining the search terms used by Twint it could be tuned to successfully extract tweets about UK politics. The final search term was as follows:

"UK AND (Labour OR Conservative OR Parliament OR Government OR "Lib OR dems" OR "liberal OR democrats") min_faves:100"

The search term looked for tweets that had the word/hashtag "UK" and any of the subsequent terms. An extra search term "min_faves" was also added to limit the returned tweets to a minimum of 100 likes. This was done to try to filter out tweets that could be perceived as "noise", whether these be from spam accounts or radical tweets that no one agrees with. By adding this criteria the authors hoped to remove the influence of the individual and instead place an emphasis on tweets that captured the opinions of the population. It should be noted, that this value was not empirically decided upon and could be open to further tuning. The search left a data set of ~37,000 tweets to perform sentiment analysis with.

VADER sentiment analysis [8] was used to embed the content of tweets into a numerical value for the sentiment of the tweet. VADER is an incredibly helpful tool designed specifically for sentiment analysis in a social media context thus no cleaning of the data was used. Lemmatisation, removal of stopwords, emojis and punctuation all affect the performance of VADER adversely and as such none of these pre-processing steps are required. The mean sentiment for all tweets in each month was calculated to give a value from each month in the time frame, a graph of which can be seen in Figure 1. It can be seen in Figure 1 that some normalisation was done to the dataset. This was done because, in the early days of Twitter, very few tweets would get more than 100 likes. As a result, the average sentiment was heavily impacted by the tweets of very few individuals and had a high variance as seen in the blue line in Figure 1. To address this issue, the average sentiment was scaled with the following formula:

$$(\frac{-1}{0.3 \cdot x + 1}) + 1$$

The data after scaling can be seen in the orange line in Figure 1.

### Target variable

The authors selected the price-earnings ratio (PE) as the target variable during the analysis of the dataset. This is due to its

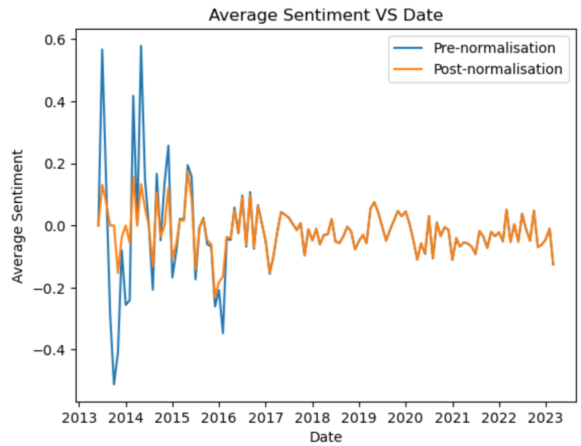


**Figure 1. A graph showing the average sentiment before (blue) and after normalisation (orange).**

ability to capture the market opinion of the luxury fashion industry by factoring in the share price as well as the actual performance of the company by accounting for the company earnings. PE is calculated with the following formula:

$$PE = \frac{SharePrice}{EarningsPerShare}$$

The data utilized for the PE, used as the target variable [3] is an index of 8 large fashion companies from the UK.

## DATA PREPARATION AND FEATURE ENGINEERING

A diagram showing the overall process for feature preparation can be seen in Figure 3.

To begin data preparation, all features were de-meaned and normalised by applying the following transformation:

$$x_i \in feature_\alpha, x_i = \frac{x_i - \mu_{feature_\alpha}}{\sigma_{feature_\alpha}}$$

Normalised features are an important property when dealing with any dataset that has features with large variations in magnitude. In the case of the dataset used in this study, some features (such as GDP and sentiment) are different by about 9 orders of magnitude. Features are normalised to ensure the learning machine avoids placing more importance on some features than others due to their magnitude.

Many statistical models used for time-series forecasting assume that the data used for the features is stationary. Data is only considered stationary if statistical properties (for example the mean and variance) of the data remain constant over time. Stationarity means that the central limit theorem can be applied to data and also opens up avenues for spectral analysis. This can be used to decompose a time series into different frequencies to analyse trends on different time scales, an example of this is Meta's Prophet [18] which uses Fourier analysis to sense trends on different time scales. To ensure

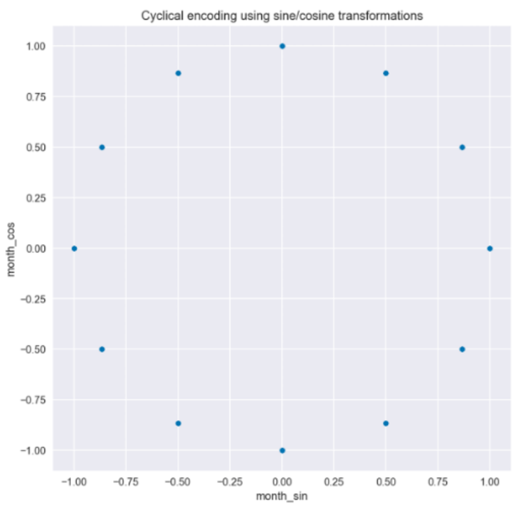


**Figure 2. A plot showing the shape of monthly data after applying the cyclical embedding. Source: Nvidia [1].**

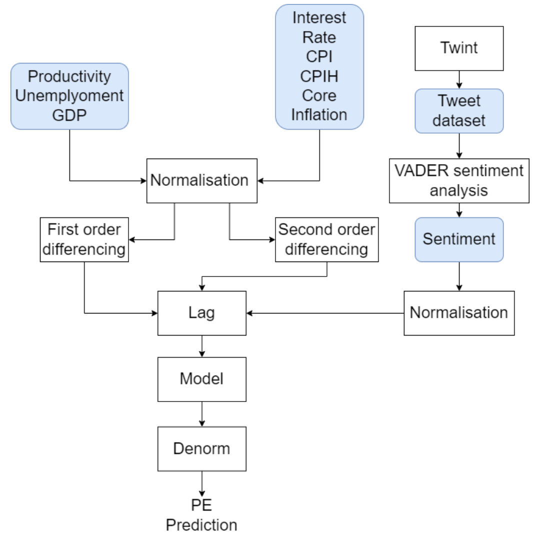


**Figure 3. The workflow for preparing the data for use in our models.**

the different features in the data set were stationary, the Augmented Dicky-Fuller (ADF) test [7] was used. The ADF tests are conducted to assess the null hypothesis of the presence of a unit root in the data. Given a low enough p-value (0.05 in this study) the null hypothesis can be rejected and data can be assumed to be stationary. If data is not stationary there are two options presented: differencing or a log-norm transformation.

After investigating the log-norm transformation the authors found that not all features were made stationary by the transformation. As such, it was decided to keep the type of transformation consistent between features, therefore, first/second order differencing was used to make features stationary.

Another form of feature engineering that was used was the embedding of monthly data in a cyclical fashion. Each entry in the dataset has a month and year value. Leaving the year value encoded as an integer makes sense, as a model should perceive that 2014 and 2016 are two years apart. However, encoding months with an integer does not make sense. On an integer number line January is encoded as 1 and December is encoded as 12. To a model this makes December 2014 and January 2015 seem far away from one another, but in reality, they are quite close. A solution to this problem is to cyclically embed the month using sine and cosine transforms. A plot showing the outcome of this transformation can be seen in Figure 2. As we can see the number line is transformed into a circular shape, allowing a model to capture the temporal adjacency of January and December.

Finally, each feature had a lag of 1-12 months added to it to create a new feature. This was done because it may take a while for a change in one feature to make itself apparent in the target variable.

## EXPLORATORY DATA ANALYSIS (EDA)

Spurious feature selection can often lead to spurious rule learning in models. As a result, it is important to perform EDA on the newly formed dataset to identify the most important features. This is an essential step in finding the right trade-off between model complexity and generality and therefore finding the optimal learning machine for the bias-variance dilemma.

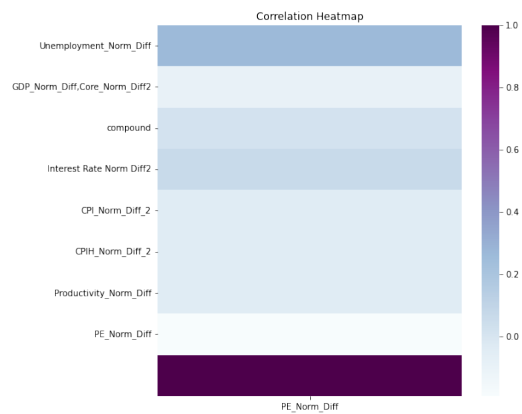


**Figure 4. A heatmap showing the correlation between various features and the target variable, PE. A darker colour indicates a stronger positive correlation, white indicates a negative correlation.**

A heatmap displaying the correlation between various features and the target feature can be seen in Figure 4. This heatmap was used to identify the most important features so they could be added to models first. Despite the importance of features is not solely determined by correlation, this graph serves as a useful starting point that provides a preliminary understanding of the features in the dataset.

After identifying the features with the strongest correlation, further testing was performed to find the lag from that feature that resulted in the strongest correlation. Following a similar method to the one mentioned before, the results showed that lags of 2, 1, 7, 9, 12, 2, 2 and 3 months were the best options for unemployment, core inflation, GDP, sentiment, interest rate, CPI, CPIH and productivity respectively.

## MODEL TRAINING AND HYPERPARAMETER TUNING

### Naive Baseline

In this analysis, we have used three different naive baseline methods to forecast the PE values from a given dataset. The naive baseline methods are often used in time series forecasting as a simple way to make predictions [4] and serve as a benchmark to compare more advanced models against. For each of these methods, we calculated the Mean Absolute Percentage Error (MAPE) over the last twelve months of predictions, the results can be seen in table 1. MAPE is a popular metric for forecasting accuracy which expresses the average absolute percentage error relative to the true value. [6]. MAPE provides a standardized measurement of prediction error. It captures the average magnitude of the prediction errors relative to the actual values, allowing us to ascertain the relative effectiveness of our naive forecasting methods [9, 12].

**Table 1. MAPE values for different models**

| Method | MAPE (%) |
|---|---|
| Last Observation Carried Forward (LOCF) | 104.64 |
| Simple Moving Average (SMA) | 95.93 |
| Exponential Moving Average (EMA) | 97.28 |
| XGBoost | 157 |
| Monte Carlo Simulation | 22.6 |
| Prophet | 11.36 |

Comparing the naive baselines, the Simple Moving Average (SMA) approach demonstrated superior performance, based on it's small MAPE value. The SMA method, predicated upon the calculation of an unweighted mean of a preceding set of observations ('n'), has evidenced a certain efficacy in mitigating statistical noise and encapsulating trends across a predefined interval. Although the SMA method may be critiqued for its potential latency in response to emergent shifts, its performance in this instance has surpassed that of the Exponential Moving Average (EMA).

The Exponential Moving Average (EMA) method, would have been expected to demonstrate superior performance due its weighting scheme that amplifies the significance of recent observations, thus facilitating a swift adaptation to data volatility, but this did not manifest the expected level of performance in this specific scenario.

Finally, the Last Observation Carried Forward (LOCF) method showed the worst performance. The LOCF method makes the simple assumption that the future value of a variable will be identical to its current value. This assumption of data stability may hold true in certain situations where there is little to no trend or seasonality. However, in the context of our dataset, which likely contains trends or seasonality given its economic nature, this assumption may fall short.

It is crucial to note that while naive methods provide a good starting point for time series forecasting, more sophisticated methods offer superior performance in capturing complex data patterns [15].

### Prophet

Prophet is a time-series forecasting model developed by Facebook (now Meta). Prophet has a range of features that make

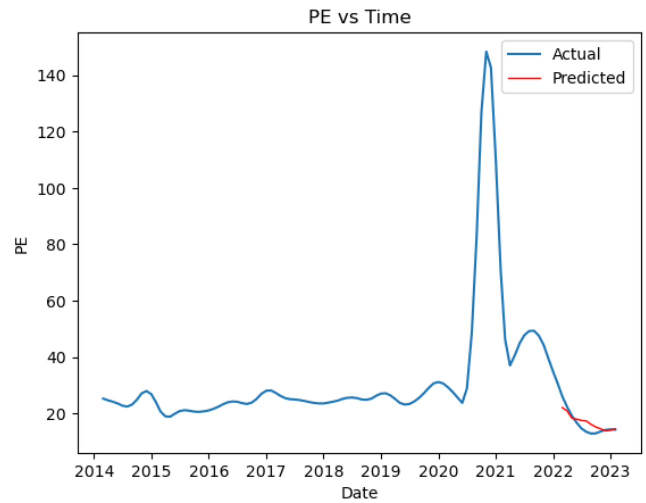


**Figure 5. A plot showing the prediction of the best performing prophet model (MAPE = 11.36%).**

it an incredibly powerful time-series modelling tool, such as seasonality modelling and automatic parameter selection. It also has the functionality to specify holidays which can be useful for preventing short-term shocks wildly affecting the model during training. Prophet can be used as a uni-variate time-series prediction model, so the model was initially trained using just the target variable as the training data. This gave a MAPE of 57.65%. It's easy to recognise how COVID-19 would wildly affect the training of the model. The Prophet documentation has steps on how to deal with COVID-19; recommending the use of the built-in holiday function to identify the data as abnormal [14]. Using this information, the dates 2020-03 to 2021-03 were identified as the peak impact period of COVID and configured as holidays in the model. Retraining the model including this parameter caused the MAPE to drop down to 26.71%. Finally, features were added to increase model complexity and allow fitting to the uncertain period post-COVID. Although important features and their corresponding lags had been identified in the EDA phase of the project, it was decided to train Prophet repeatedly using a brute-force method of all feature variations to find the best-performing features specific to Prophet. Given the fact that it only takes around 15 seconds to train a new model, this problem is tractable to brute force and only took a few hours. The best-performing model only used the following 3 features: Productivity lag 5, GDP lag 7 and Interest Rate lag 3 and achieved a MAPE of 11.36%. A plot of this model's prediction can be seen in Figure 5.

### XGBoost

XGBoost, or 'Extreme Gradient Boosting', is a gradient-boosted decision tree algorithm. It is often used for regression, classification and ranking problems in machine learning applications [16]. Furthermore, it has been successfully applied to forecasting problems with time series data, achieving a short-term prediction with MAPE $< 5\%$ in traffic forecasting for telecommunications applications [10].

To find the best hyperparameters, the authors ran a grid search to find the combination of hyperparameters that provided the lowest MAPE on the test split of the dataset. The documentation for the XGB Regressor shows that we can alter the number of gradient-boosted trees, $\beta_1$, the maximum tree depth for base learners, $\beta_2$, and the learning rate, $\gamma$. We run experiments over the following sets:

$\beta_1 = \{50, 75, 100, 250, 500, 750, 1000\}$
$\beta_2 = \{3, 5, 7, 9, 11, 15\}$
$\gamma = \{0.01, 0.05, 0.1, 0.2, 0.3\}$

Through this approach, we found the best hyperparameters as $\beta_1 = 75$, $\beta_2 = 11$, $\gamma = 0.2$.

After tuning the hyperparameters, we investigated the best predictions for the final 3 months of *PE* as compared with the test split. We found that we can achieve a minimum MAPE of 157%.

By using the model to predict over a longer time span (that covers 20% of the time covered by the whole dataset) we found that the model may be biased to pre-COVID patterns. Due to the effects of COVID, the *PE* volatility increases, resulting in the XGBoost algorithm opting to jump into predicting patterns similar to the data from before the effects of COVID, as shown in Figure 6. This may be affecting our predictions on the final 3 months of the dataset and may mean this report is not a fair representation of the performance of the XGBoost model on time series predictions in a different economic climate.

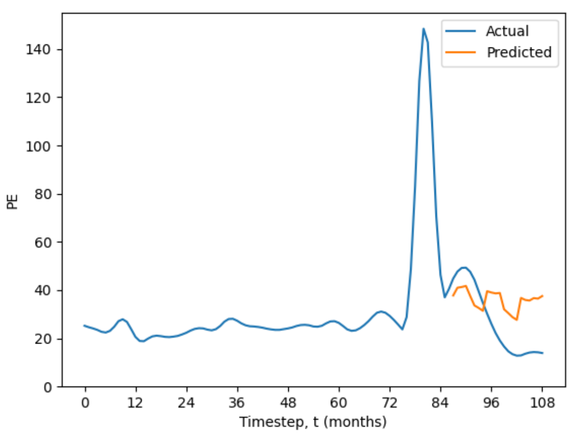


**Figure 6. XGB predictions over a longer time span. We can see the effect of COVID on the target *PE*, and how the model handles prediction after this event.**

## Monte Carlo Simulation

Monte Carlo Simulation is a method that can be used to forecast the value of financial assets. It works by performing repeated stochastic computational simulations [11], taking in historical data alongside a random distribution, such as the normal distribution, as inputs to provide projected results. Typically, when used in a financial context, the input features are different stocks and the output value will be the expected portfolio value, calculated by generating cumulative daily returns and adding that to a starting value. In our implementation, the method for generating the stochastic forecasts involves assuming the monthly change in the PE value, $R$, follows a multivariate normal distribution,

$$R \sim N(\mu, \Sigma)$$

Cholesky Decomposition yields a Lower Triangle Matrix, $\mathbf{L}$, which can be used to correlate the random samples from the normal distribution with the covariance matrix, $\Sigma$, of our features [5], where

$$\mathbf{L}\mathbf{L}^{\intercal} = \Sigma$$

The monthly returns are generated by taking the mean of the historical monthly percentage changes in PE and adding them to the dot product of the $\mathbf{L}$ matrix and the uncorrelated random variables. Finally, the cumulative monthly changes are recorded per simulation.

Testing was undertaken to determine which features provided the most accurate simulations, by using the MAPE of the prediction compared to the test data. The best features were CPI, GDP, Core Inflation and Productivity which, after the weights are optimised, yield a MAPE of 22.6%. The results when using these features are shown in Figure 7. It should be noted that many features were not suitable for this method as a reasonable correlation was needed between the monthly change in the feature's value and the change in the PE value.

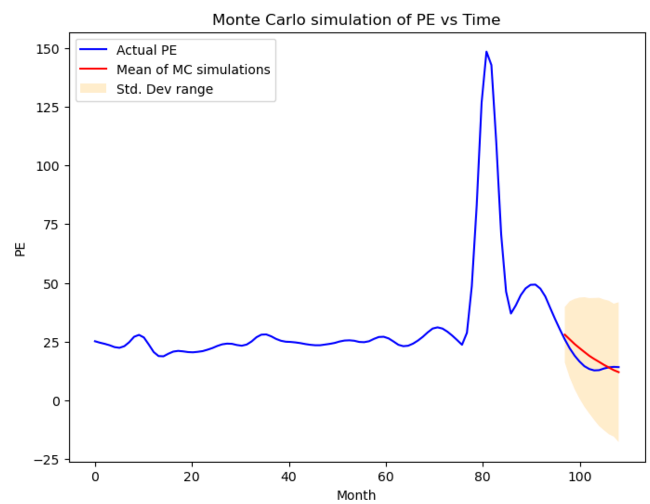


**Figure 7. Monte Carlo simulation result, forecasting the value of PE over 12 months based on the mean of 50000 simulations. The orange shaded area represents a range of one $\sigma$ each direction. MAPE=22.6%**

## CONCLUSION & REFLECTION

Overall, the best performing model was clearly Prophet, achieving a MAPE of 11.36%. To make further improvements it would be worth investigating changing the sampling frequency for all of the data to quarterly and extending the time frame from 2013-2023 to 1983-2023. This would have resulted in the same number of data points but would have the upside of the vast majority of data points being actual data points rather than data from interpolation.

Some of the models exhibited improved performance by making data points during covid less important (Prophet) or removing the data points altogether. In hindsight, this was a rather myopic decision because it removes important information that could help model performance post-COVID. All models struggle to capture the behaviour of the PE after COVID because it is unlike anything seen in prior data. This could be combated by giving the model a feature to sense the severity of COVID such as the UK stringency index [13].

The datasets also lack any indicators of the global macroeconomic state. All of the stocks used in the index for the target variable are large Multi-National Corporations (MNCs) that would be affected by customer sentiment all over the world. This could in turn be affected by things such as the Ukraine war.

Unfortunately, in the hyperparameter tuning stage of Prophet, the sentiment feature was not included as part of the best-performing combination of features. It should be mentioned that it had a strong performance as the sole regressor of prophet, achieving a MAPE of 19.93%. In hindsight, the sentiment dataset could've been improved in the data collection phase by removing tweets from official party outlets because these would always be biased towards the party creating the tweet.

In conclusion, this report details the successful creation and use of a dataset that allows for the prediction of the economic state of the UK luxury fashion industry using national macroeconomic indicators.

## REFERENCES


[1] 2022. Nvidia cyclical encoding. https://shorturl.at/kARTZ. (2022). Accessed: 16.05.2023.

[2] 2023. Organisation for Economic Co-operation and Development. https://data.oecd.org/interest/long-term-interest-rates.htm. (2023). Accessed: 16.05.2023.

[3] 2023. Simply Wallstreet. https://simplywall.st/markets/gb/consumer-discretionary/luxury. (2023). Accessed: 16.05.2023.

[4] J. S. Armstrong. 2001. *Principles of Forecasting: A Handbook for Researchers and Practitioners*. Springer Science & Business Media.

[5] Nicholas Burgess. 2022. Correlated Monte Carlo Simulation using Cholesky Decomposition. *Available at SSRN* (2022).

[6] T. Chai and R. R. Draxler. 2014. Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. *Geoscientific Model Development* 7, 3 (2014), 1247–1250.

[7] Yin-Wong Cheung and Kon S Lai. 1995. Lag order and critical values of the augmented Dickey–Fuller test. *Journal of Business & Economic Statistics* 13, 3 (1995), 277–280.

[8] C.J. Hutto. 2023. VADER sentiment analysis. https://github.com/cjhutto/vaderSentiment. (2023). Accessed: 16.05.2023.

[9] R. J. Hyndman and G. Athanasopoulos. 2018. Forecasting: Principles and Practice. (2018). https://otexts.com/fpp3/

[10] Garima Jain and Rajeev Ranjan Prasad. 2020. Machine learning, Prophet and XGBoost algorithm: Analysis of Traffic Forecasting in Telecom Networks with time series data. In *2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*. 893–897. DOI: http://dx.doi.org/10.1109/ICRITO48877.2020.9197864

[11] Belva Jones and Gerald E. Evans. 2009. The Application of Monte Carlo Simulation in Finance, Economics and Operations Management. (2009). DOI: http://dx.doi.org/10.1109/CSIE.2009.703

[12] B. Makridakis. 1984. The forecasting accuracy of major time series methods. *Journal of the Royal Statistical Society: Series A (General)* 147, 2 (1984), 145–162.

[13] Edouard Mathieu, Hannah Ritchie, Lucas Rodés-Guirao, Cameron Appel, Charlie Giattino, Joe Hasell, Bobbie Macdonald, Saloni Dattani, Diana Beltekian, Esteban Ortiz-Ospina, and Max Roser. 2020. Coronavirus Pandemic (COVID-19). *Our World in Data* (2020). https://ourworldindata.org/coronavirus.

[14] Meta. 2023. Prophet documentation on handling the covid shock. https://facebook.github.io/prophet/docs/handling_shocks.html. (2023). Accessed: 16.05.2023.

[15] D. C. Montgomery, C. L. Jennings, and M. Kulahci. 2015. *Introduction to Time Series Analysis and Forecasting*. John Wiley & Sons.

[16] NVIDIA. XGBoost - What Is It and Why Does It Matter? (????). https://www.nvidia.com/en-us/glossary/data-science/xgboost/

[17] Francesco Poldi. 2023. Twint. https://github.com/twintproject/twint. (2023). Accessed: 16.05.2023.

[18] Sean Taylor and Benjamin Letham. 2017. Forecasting at Scale. *The American Statistician* 72 (09 2017). DOI: http://dx.doi.org/10.1080/00031305.2017.1380080