Demand Prediction for the Santander Bikes across London

Candidate 26527, Candidate 24738, Candidate 25411, Candidate 34553

ABSTRACT

This study aims to predict demand and revenue for Santander cycles by analyzing historical usage data and related factors. The analysis integrates complex datasets, including cycling patterns and weather conditions, using advanced statistical and machine learning models such as Linear Regression, Gradient Boosting Trees (GBT), and Random Forests. These models were employed to predict trip duration, which is further used to derive revenue prediction. The study also explores the proximity of bus and tube stations as a peripheral factor to assess its impact on cycle usage, incorporating geospatial matching with bus stop and tube station data using BigQuery GIS. Revenue forecasting model was trained on ARIMA, SARIMA and LSTM models. The final models chosen for the combined models included the GBT and SARIMA models, which outperformed the other models with final RMSE values of 435.06 and 321.350 respectively.

Keywords: Demand Forecasting, Revenue Prediction, Linear Regression, Gradient Boosting Trees, Random Forest.

1 INTRODUCTION

Santander Bikes have become a cornerstone of commuting in London, enjoying widespread usage among its residents. Understanding the demand for these bikes and the resulting revenue is essential for making informed decisions. This is also important to help Santander structure their cycle stops and place right amount of bikes to reduce their energy waste by maximising their revenue. Reliable prediction is also necessary to push the country for a more carbon-neutral economy, by catering to the demand of as many people as Santander can and promoting a greener mode of commute. This research paper endeavors to develop a machine learning model tailored for big data-friendly revenue prediction of Santander Bikes across the UK.

The primary goal of this study is to build a model that is better able to predict the demand for Santander bikes across London and then forecast their revenue. To achieve this, we use advanced predictive modeling techniques using Python within the PySpark framework. Our innovative approach involves creating a hybrid model that integrates two key components: predicting the total duration of bike rentals and subsequently forecasting revenue using time series analysis. We constructed three models for demand prediction - Linear Regression Model, Random Forest and Gradient Boosted Tree model. Further, for the revenue prediction, ARIMA - Autoregressive Integrated Moving Average model, SARIMA - Seasonal Autoregressive Integrated Moving Average model and LSTM - Long Short term memory network models were used.

2 LITERATURE REVIEW

The existing research papers explore multiple models to conduct a more accurate bike-sharing demand prediction. Like the research paper [7], explores ARIMA, RandomForest, CatBoost model and LSTM model. In their conclusion, they found that the CatBoost model resulted in the lowest RMSE for the data. However, LSTM model performed better for nest three-day prediction better due to its better ability to capture trends over a longer period of time. Existing studies have predicted demand for bikes through statistical methodologies, but few have applied machine learning methodologies, which have recently been in the spotlight [8].

From the perspective of bike-sharing service, the amount of rentals is a key performance indicator for managers and supervisors in demand assessment. Therefore, the prediction of bicycle demand by bike-sharing system is a key index in economic system [11]. Some of the research papers also highlight the importance of choosing appropriate features that affect the bike-sharing demand. The historical demand information was used to develop several autoregressive integrated moving average (ARIMA) models by using Box–Jenkins time series procedure and the adequate model was selected according to four performance criteria [4]. The auto-ARIMA model was used to pick the initial variations of the model parameters and then the optimal model parameters were found based on the best match between the forecasts and test data. The models reliability was evaluated using the analytical methods Auto Correlation Function (ACF), Partial Auto-Correlation function (PACF), Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC), [6].

3 DATA

The data used for building the models are taken from Google Cloud Platform's database and the data is called London Bicycle. This dataset contains around 83 million rows of data and is a total of 9.57 GB. Due to computational limitations, time limitations, and limited GCP credit availability, we have used the data from years 2019 to 2020 for predictions. The link to the dataset can be found in the references. The below snapshot explains the type of the variable for each. The data contains information about the start station, time of the rental and the end station and time along with the date and rental id.

rental_id	duration	duration_ms	bike_id	bike_model	end_date	end_station_id	end_station_name	start_date	start_station_id	start_station_name	end_station_logical_terminal	start_station_logical_terminal	end_station_priority_id
99237458	1800	1800000	11064	null	2020-07-09 21:59:00	468	Cantrell Road, Bow	2020-07-09 21:29:00	256	Houghton Street, ...	null	null	null
102119456	780	780000	7021	null	2020-09-20 11:52:00	245	Grosvenor Road, P...	2020-09-20 11:39:00	256	Houghton Street, ...	null	null	null
91191594	780	780000	2788	null	2019-09-13 14:30:00	340	Bank of England M...	2019-09-13 14:17:00	256	Houghton Street, ...	null	null	null
102328665	780	780000	18356	null	2020-09-25 16:18:00	26	Ampton Street , C...	2020-09-25 16:05:00	256	Houghton Street, ...	null	null	null
101663526	780	780000	15827	null	2020-09-09 20:07:00	351	Macclesfield Rd, ...	2020-09-09 19:54:00	256	Houghton Street, ...	null	null	null

3.1 Public Holidays

To make the analysis more comprehensive, we have scrapped the public holidays data for the years 2019 to 2022 from publicholidayguide.com. This dataset was pre-processed for a model ready format, by removing unnecessary characters from the dates, adding years to the dates and so on.

3.2 Weather Data

To improve our bike rental demand prediction models, we extracted extensive weather data from 2019 to 2022 using an API from visualcrossing.com. Since the weather data was available for every

hour of the day, the entire cycle hires data was aggregated for every hour. This allowed a seasless integration of the weather data with our cycle hire records through an inner join on datetime attributes, aligning hourly weather conditions such as temperature, humidity, precipitation, and windspeed with corresponding rental events. We standardized datetime formats across both datasets and converted rental durations from milliseconds to seconds for uniform analysis. Redundant columns were removed to focus on essential predictive variables. These preprocessing steps ensured that the dataset was not only streamlined but also enhanced to include critical environmental factors, establishing a robust foundation for accurate demand forecasting.

3.3 Bus and Tube data

In addition to the dataset from BigQuery "London Bicycles", data on bus stops and tube stations were sourced from the Transport for London (TfL) open data portal. These datasets were integrated using Apache Spark, facilitated by a SparkSession that allowed for efficient querying and data manipulation.

4 EXPLORATORY DATA ANALYSIS

The cycle hires data was used to conduct a simple Chisquare test to check whether there was statistical evidence to say that there is a relation between the start stations and the end stations for each rental. The Chisquare test was conducted with the below hypothesis

-

- H_0: The start stations and the end stations for every ride are independent.
- H_1: The start stations and the end stations for every ride are not independent.

The Figure 21 shows that there is a relation between the start station and the end station.

The analysis of bus stops and tube stations data began with the establishment of a comprehensive database in BigQuery, which was selected for its robust handling of large-scale geospatial data. The initial phase involved loading the bus stops, tube stations, and cycle hire datasets into BigQuery. These datasets, each with distinct structures, required meticulous integration to ensure uniformity for subsequent queries. Once ingested, the data underwent a geospatial matching process utilizing BigQuery's GIS capabilities. A 100-meter radius was defined around each cycle hire station to locate nearby bus stops and tube stations within this proximity threshold. The `ST_DWITHIN` function enabled precise spatial comparisons between the cycle hire locations and surrounding transport facilities. To perform geospatial matching of cycle hires with the bus stops and tube stations, BigQuery GIS capabilities were used to identify bus stops and tube stations within a 100-meter radius of each cycle hire station. This geospatial analysis enabled the researchers to assess the proximity's impact on cycle hire demand. The resultant datasets were then analyzed to explore the popularity of bus stops and tube stations based on their proximity to cycle hire locations. Furthermore, exploratory data analysis (EDA) was conducted to examine the correlation between the nearest distance to bus and tube stations and the demand for cycle hires. This approach provided insights into how the accessibility of public transportation influences cycle hire patterns, contributing to a more nuanced understanding of urban mobility in London. The distribution of cycle stands across London, illustrated in Figure 1, shows a pronounced concentration in the central region suggesting higher utilization in dense urban areas.

Figure 1: Cycle stands in London

Figure 2 provides a comprehensive view of cycle stands, bus stops, and tube stations, demonstrating the interconnectivity of London's transport network and its extensive coverage.

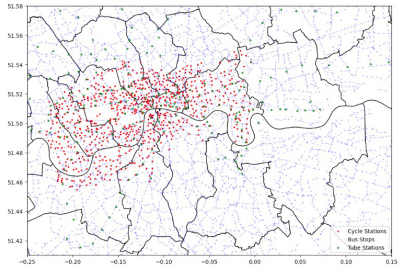

Figure 2: Cycle stands, Bus stops and Tube stations in London

The map in Figure 3 highlights the top 20 bus stops by cycle hire frequency, indicating popular nodes of travel within the city's transport matrix.

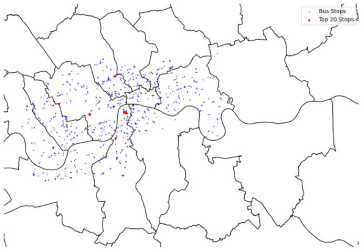

Figure 3: Top 20 Bus stops by cycle hires

Similarly, Figure 4 presents the most frequented tube stations by cycle hire volume, showcasing the hotspots of commuter activity and intermodal connectivity.

For modelling purposes, this research paper uses data aggregated from the cycle hires along with the weather data. A correlation plot between these variables in Figure 5.

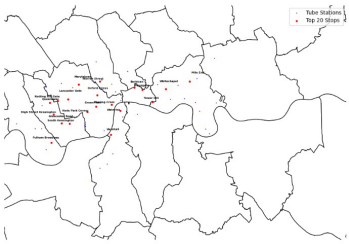

Figure 4: Top 20 Tube stations by cycle hires

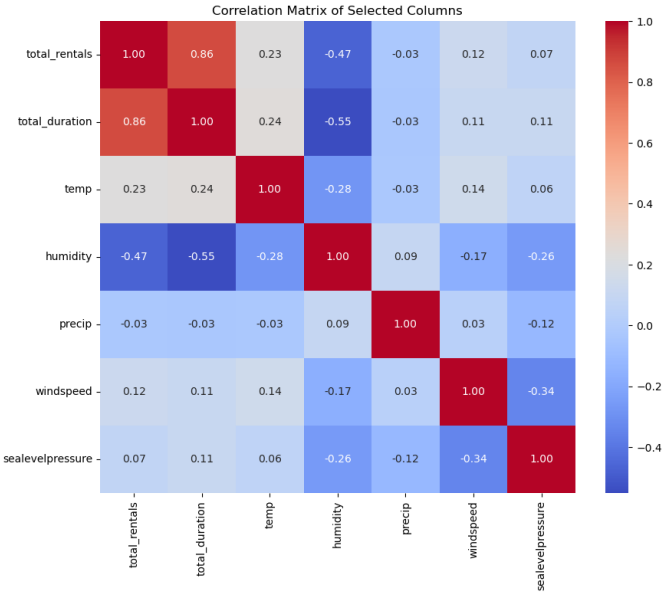

Figure 5: Correlation plot

5 DEMAND PREDICTION

The primary objective is to forecast the total demand in the context of bike rentals by predicting the total duration of rentals. This task is tackled as a regression problem where the dependent variable y is the 'total duration' of bike rentals and the independent variables X include:

- X_1 `public_holiday_Vec`
- X_2 `day_of_week_Vec`
- X_3 `temp`
- X_4 `precip`
- X_5 `windspeed`
- X_6 `sealevelpressure`
- X_7 `total_rentals`
- X_8 `datetime`

5.1 Linear Regression Model

The Linear Regression model is predicated on the assumption of a linear relationship between the dependent variable and multiple independent variables. The relationship can be mathematically represented as follows:

$$y = \beta_0 + \sum_{i=1}^{n} \beta_i x_i + \epsilon \tag{1}$$

where:

- β_0 is the intercept term.
- $\beta_1, \ldots, \beta_n$ are the coefficients representing the effect of each independent variable x_i on the dependent variable y.
- ϵ is the error term, accounting for the variability in y that cannot be explained by the linear relationship with x_i.

This model not only serves as a fundamental analytical tool in both statistics and machine learning but also acts as a baseline for comparing the performance of more complex algorithms. To combat overfitting, especially prevalent in high-dimensional data scenarios, a regularization parameter of $\lambda = 0.01$ is employed. The maxIter is set to 100 for the model to specify that the optimization algorithm should iterate up to 100 times to find the best model parameters. This helps in achieving convergence to the optimal solution, while balancing computational cost and model accuracy.

5.2 Gradient Boosted Trees (GBT) Model

Gradient Boosting Machines (GBMs) are powerful ensemble learning techniques that sequentially fit new models to improve the accuracy of predictions. The fundamental idea behind GBMs is to construct base learners that are highly correlated with the negative gradient of the loss function associated with the entire ensemble. This iterative process results in the gradual reduction of the residual errors in the predictions. [9]

5.2.1 *Model Parameters.*

- **maxDepth**: This parameter is set to 5, which means that each tree in the ensemble can have a maximum depth of 5 levels. This controls the complexity of each individual tree.
- **maxIter**: This parameter is set to 100, indicating that the maximum number of iterations (or trees) in the ensemble is 100. Each iteration adds a new tree to the ensemble.
- **stepSize**: This parameter is set to 0.1, which determines the contribution of each tree to the overall ensemble. A smaller step size makes the model more conservative and less prone to overfitting.

The equation for the GBT Regressor model is given by:

$$\mathrm{y} = \sum_{i=1}^{N} \eta \times F_i(x) \tag{2}$$

Where:

$$F_i(x) = \begin{cases} 0 & \text{if tree } i \text{ is not applicable for input } x \\ \text{prediction of tree } i & \text{otherwise} \end{cases}$$

The GBT model begins with a simple guess, usually the average value of the target. Each new tree learns to correct the mistakes made by the previous ones, aiming to get closer to the actual outcomes. In our model, we set a limit of 5 for how deep these trees can go (MaxDepth parameter), which helps keep things manageable.

The amount each new tree contributes to the final prediction is decided by a process that tries to minimize mistakes, overall. We

set the learning rate to 0.1 (StepSize parameter), which controls how much each tree's guess affects the final prediction.

In the end, the model combines all these individual guesses to make its final prediction. This way of learning from past mistakes helps the model deal with complex relationships in the data.

5.3 Random Forest Model

The Random Forest model is an ensemble learning technique that builds upon the simplicity of decision trees by creating a forest of them. During the training phase, the Random Forest algorithm constructs multiple decision trees, each trained on a random subset of the training data [2]. The general formula for prediction in a Random Forest is given by:

$$y = \frac{1}{B}\sum_{b=1}^{B} h_b(x) \tag{3}$$

where B denotes the total number of trees in the forest and $h_b(x)$ represents the prediction of the b-th tree. This structure allows the model to harness the strengths of multiple decision trees, which are averaged to produce a more accurate and stable prediction. The averaging process effectively reduces the variance without increasing bias, which mitigates the risk of overfitting that is typical in a single decision tree.

In practical implementations of a Random Forest, key parameters such as the number of trees (B) and the depth of each tree are crucial for balancing the model's accuracy and computational efficiency.

5.3.1 *Model Parameters.*

- **maxDepth**: The maximum depth of each decision tree in the ensemble. This parameter is set to 10.
- **numTrees**: The number of trees in the Random Forest ensemble. This parameter (B) is set to 50.

For this model, B is set to 50, meaning that fifty trees contribute to the final average prediction. Limiting each tree's depth to 10 helps to control overfitting while still capturing sufficient complexity in the data.

5.4 Model Pipeline

We conducted an elaborate model pipeline phase, which included the sequence mentioned in Figure 6. The flow chart shows the steps performed under data preparation, model set up, model prediction and model evaluation.

5.5 Model Evaluation

To assess the performance of our models, we use the Root Mean Square Error (RMSE), which provides a measure of the differences between values predicted by the model and the values observed:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{4}$$

where N represents the total number of data points in our test set, which is 30% of the total data after splitting, y_i are the actual durations, and $\hat{y}_i$ are the predicted durations from the model. This metric helps us understand the average error magnitude across all predictions. Each model was encapsulated within a PySpark pipeline, allowing for seamless data transformation and model application.

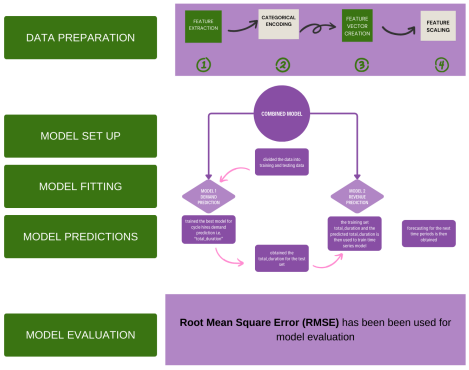

Figure 6: Model

5.6 Results

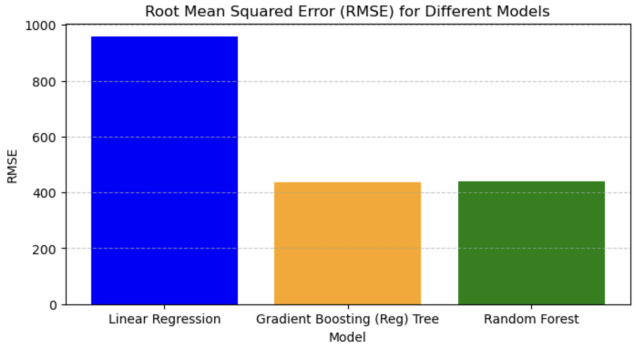

Figure 7: RMSE comparison

In our comparison of predictive models, the Gradient Boosted Trees (GBT) model stood out as the best performer, with a Root Mean Square Error (RMSE) of 435.066. This is a significant improvement over the Linear Regression and Random Forest models, which had RMSE values of 957.212 and 440.095, respectively. The GBT model's ability to capture complex relationships in the data led to its superior performance.

The GBT model excels in identifying important features for prediction, such as 'total rentals' and 'year'. Its iterative error-correction approach, where each new tree improves on the predictions of the previous ones, further enhances its accuracy. The model's flexibility in handling various predictors, especially in datasets with time-series elements and diverse weather variables, makes it a reliable choice for predicting bike rental demand.

The feature importance plot (refer to Figure 8) reveals insightful comparisons between the Gradient Boosted Trees (GBT) and Random Forest models. Notably, the GBT model demonstrates superior performance across most metrics, underscoring its effectiveness in capturing complex relationships within the data. However, it's noteworthy that the Random Forest model assigns higher importance to

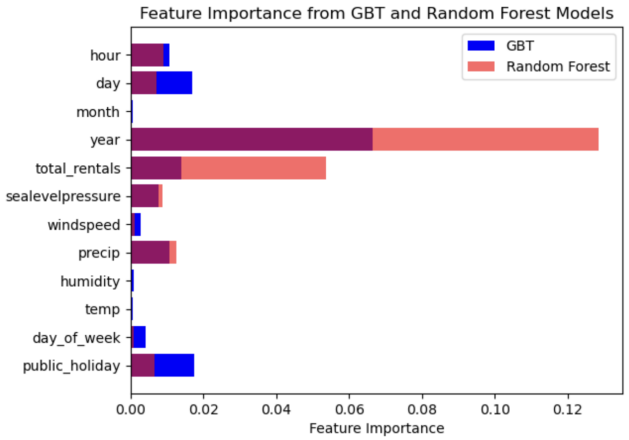

Figure 8: Feature Importance.

Note: The maroon colour indicates overlapping values in GBT and Random Forest models.

the 'year' and 'total_rentals' variables compared to the GBT model. This discrepancy suggests that the Random Forest trees might comparatively rely on these specific variables more, potentially leading to overfitting and limited generalization on unseen data.

6 REVENUE ANALYSIS AND FORECASTING

Based on the analysis conducted above, we conclude that the Gradient Boosted Tree model demonstrates the highest accuracy in predicting the demand for the total duration of rides on an hourly basis. With this duration information, our next objective is to analyze the revenue generated for the years 2019 and 2020 using different time-series models. Additionally, we aim to forecast the revenue for the next 100 days.

To estimate the revenue, we make the following assumptions:

(1) All bikes used are standard bicycles (not e-bikes).
(2) There are no monthly members, and all rides are paid individually.
(3) We assume that the pricing structure remained constant over the two years under consideration, as specific information regarding trip types and prices is unavailable.

Payment plans:

- Base price = £1.65 for the first 30 mins.
- Additional charge = £1.65 for each additional 30 minutes.

For more details, refer to Transport for London's page on Santander Cycles pricing [12].

6.1 Data Preprocessing

First and foremost, we create a function *calculate_revenue* to compute the revenue for both the actual and predicted revenue for every individual ride based on the provided payment plan. We then aggregate the revenue on a daily basis to facilitate further analysis.

Before proceeding with the time series analysis, we conduct a two-sample z-test to assess whether the distributions of the actual and predicted revenue per day, based on the Gradient Boosted Tree (GBT) model, exhibit statistically significant differences.

6.1.1 ***Two-Sampled Z Test.*** The two-sample z-test is employed when we have two independent groups with continuous normally distributed outcomes. We compare their means using a z-test if the sample size is sufficiently large (i.e., $n > 30$) and the population standard error is known [10].

We define the null and alternative hypotheses as follows:

(1) H_0: The two distributions (actual and predicted revenue per day) are statistically the same.
(2) H_1: The two distributions (actual and predicted revenue per day) are statistically different.

The test statistic is computed as:

$$z = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

where:

- $\bar{x}_1$ is the mean of the actual revenue per day.
- $\bar{x}_2$ is the mean of the predicted revenue per day.
- σ_1 is the standard deviation of the actual revenue per day.
- σ_2 is the standard deviation of the predicted revenue per day.

The computed p-value is 0.999. Therefore, we cannot reject the null hypothesis, indicating that the predicted revenue per day is similar to the actual revenue per day. Consequently, we utilize the predicted revenue to proceed with various time series models.

6.2 ARIMA

The Auto-Regressive Integrated Moving Average (ARIMA) model serves as a comprehensive extension of autoregressive moving average techniques, offering invaluable insights into time-series data forecasting. ARIMA models prove particularly advantageous when confronting non-stationary mean phenomena within the data. To assess stationarity, an Augmented Dickey-Fuller test was conducted, yielding a p-value of 0.304, indicating non-stationarity. Despite this, ARIMA models excel in accommodating deterministic and periodic components, thereby facilitating robust data modeling and forecasting [5].

An ARIMA model is labeled as an ARIMA model (p, d, q), wherein:

- p is the number of autoregressive terms;
- d is the number of differences; and
- q is the number of moving averages.

The autoregressive process. The "AR" in ARIMA denotes autoregression, indicating that the model utilizes the dependent relationship between current data and its past values. In essence, it signifies that the data is regressed on its preceding values.

Autoregressive models assume that Y_t is a linear function of the preceding values and is given by equation (1):

$$Y_t = \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + \alpha_3 Y_{t-3} + \alpha_4 Y_{t-4} + \alpha_5 Y_{t-5} + \alpha_6 Y_{t-6} + \epsilon_t \quad (5)$$

This equation represents a time series model where the value of Y_t at time t is linearly dependent on its own past values up to Y_{t-6} plus an error term ϵ_t. Each term $\alpha_i Y_{t-i}$ represents the influence of the value of Y at a previous time point $t - i$, weighted by the corresponding coefficient α_i.

In our scenario, we utilize the *partial autocorrelation plot* to discover the number of autoregressive terms that might constitute the optimal model. As observed in Figure 9, autoregressive terms up to 6 display significant deviations from zero. Hence, the plausible range of p-values spans from 0 to 6.

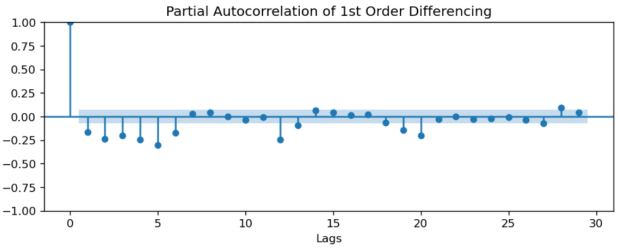

Figure 9: Partial Autocorrelation plot

The integrated process. The "I" stands for integrated, which means that the data is stationary. Stationary data refers to time-series data that's been made "stationary" by subtracting the observations from the previous values.

There is no such method that can tell us how much value of d will be optimal. However, the value of difference can be optimal till 2 so we will try our time series in both. From the figure 10 we notice that despite the original series is not stationary the first and second order differencing results in a more stationary nature, thus the possible values for d that can be used are 0,1,2

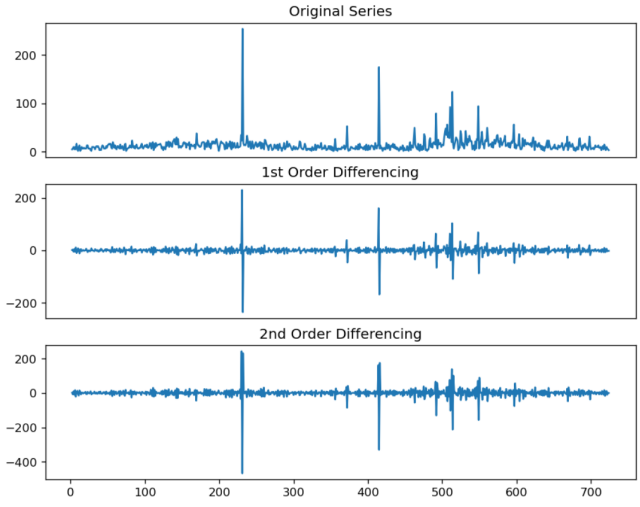

Figure 10: First and second order difference

The moving average process. The "MA" stands for moving average model, where the current value of a changing moving averaging process is a linear combination of the current disturbance with one or more previous perturbations. The moving average order indicates the number of previous periods embedded in the current value.

$$Y_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} \tag{6}$$

where Y_t denotes the value of the time series at time t, and ϵ_t represents the random and unpredictable error term at time t. The coefficients θ_1 and θ_2 indicate the influence of the lagged error terms ϵ_{t-1} and ϵ_{t-2} on Y_t, respectively.

Likewise, we notice from the Figure 11 that lags one and two are significantly different from zero, suggesting possible values for the moving average order q to be 0, 1, or 2.

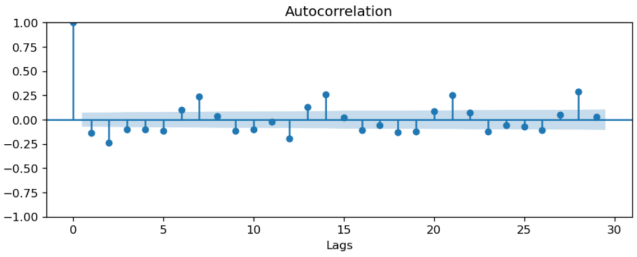

Figure 11: Auto correlation plot

6.2.1 ***Hyperparameter Tuning.*** To determine the optimal parameters, a grid search was conducted over different values of p, d, and q.

Possible values: p – [1, 2, 3, 4, 5, 6] d – [0, 1, 2] q – [1, 2]

The dataset was divided into training and testing sets, with the initial 90% of records allocated for training and the remaining 10% for testing. Root Mean Square Error (RMSE) served as the evaluation metric. The best-performing model, characterized by parameters $p = 6$, $d = 0$, $q = 2$, achieved a test RMSE of 280.501.

The equation of ARIMA (6, 0, 2) is:

$$Y_t = \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + \alpha_3 Y_{t-3} + \alpha_4 Y_{t-4} + \alpha_5 Y_{t-5} + \alpha_6 Y_{t-6} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} \tag{7}$$

6.2.2 ***Model Training.*** After determining the optimal parameters for ARIMA (ARIMA(6,0,2)), the *ARIMA* function from the *statsmodels* package is employed to train the model using the training data.

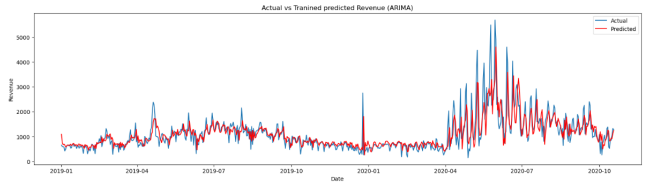

Figure 12: Training the ARIMA Model

6.2.3 ***Test Results and Forecasting.*** The trained ARIMA model is then utilized to predict values for the testing data period and the subsequent 100 days, as illustrated in Figure 13. The Root Mean Square Error (RMSE 5.5) for the testing period was calculated to be 403.691.

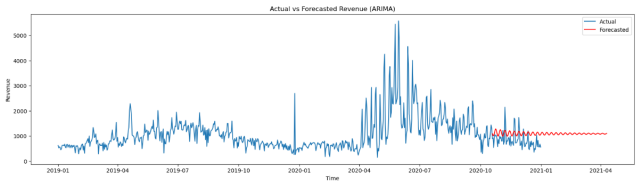

Figure 13: Actual, predicted and forecasted data (ARIMA)

6.3 SARIMA

Seasonal autoregressive integrated moving average: SARIMA is an extended algorithm that has a seasonal component along with the Auto-Regressive Integrated Moving Average (ARIMA) method. The model assumes that the revenue generated by Santander Cycles comprises trends, seasonal components, and irregular terms [3].

We can now apply another ARIMA (p, d, q) model to $\Delta D s X_t$ by multiplying the seasonal model by the new ARIMA model in order to remove any remaining seasonality and obtain a mathematical representation of SARIMA (p, d, q) (P, D, Q, S).

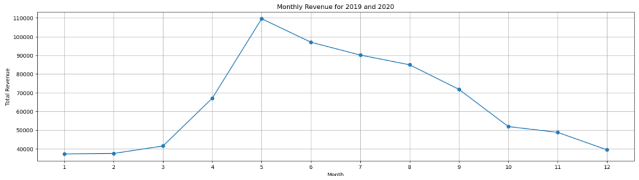

Figure 14: Monthly Trend of Revenue

From Figure 14, it can be observed that spring and summer months (April to September) exhibit higher revenue compared to autumn and winter (October to March). Therefore, in the analysis, the seasonal component is fixed to 12.

6.3.1 ***Hyperparameter Tuning.*** Hyperparameter tuning involved a grid search over a range of values for the parameters p, d, q, P, D, and Q. Due to computational constraints, S was fixed at 12. The parameter ranges used were: $p - [0, 1, 2, 3, 4, 5]$, $d - [0, 1, 2]$, $q - [0, 1, 2]$, $P - [0, 1, 2]$, $D - [0, 1, 2]$, and $Q - [0, 1, 2]$.

Similarly, the data was split into a 90:10 ratio for training and testing, with 90% used for training and 10% for testing. Root Mean Square Error (RMSE) was employed for evaluation, resulting in the best parameters being SARIMA (5, 0, 1) (0, 0, 0, 12) that achieved a test RMSE of 305.756

The equation of SARIMA (5, 0, 1) (0, 0, 0, 12) is:

$$Y_t = \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + \alpha_3 Y_{t-3} + \alpha_4 Y_{t-4} + \alpha_5 Y_{t-5} + \epsilon_t + \theta_1 \epsilon_{t-1} \quad (8)$$

6.3.2 ***Model Training.*** Similarly, for the optimal parameters of SARIMA (SARIMA(5,0,1)(0,0,0,12)), the default *SARIMAX* function from the *statsmodels* package is utilized to train the model using the training data. Refer to Figure 15 for the training process.

6.3.3 ***Testing and Forecasting.*** The same methodology is repeated for the SARIMA model by forecasting results for the testing data and the subsequent 100 days. Figure 16 demonstrates the testing and forecasting outcomes. The Root Mean Square Error (RMSE

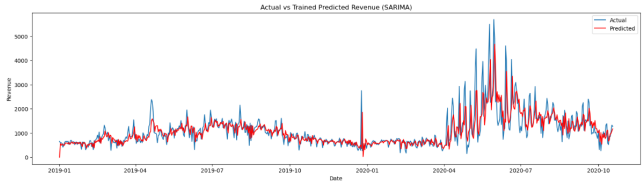

Figure 15: Actual vs Predictions on the training data (SARIMA)

5.5) between the actual and predicted results was calculated to be 321.350.

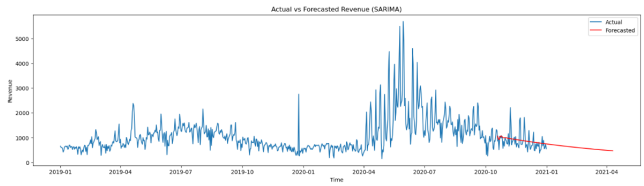

Figure 16: Actual, predicted and forecasted data (SARIMA)

6.4 LSTM

LSTM networks are tailored for capturing temporal dependencies in sequential data, making them adept at forecasting time series data. Their unique architecture enables them to retain relevant information over long periods while discarding irrelevant details, ideal for analysing data with long-range dependencies. Additionally, LSTM networks effectively address vanishing and exploding gradient issues, common challenges in training deep neural networks for sequential data. Figure 19 represents the LSTM architecture.

6.4.1 ***Model Architecture.*** The LSTM architecture employed in this case is a Vanilla LSTM, constituting a simple neural network with 1 LSTM layer and 2 Dense layers. The LSTM layer consists of 64 neurons and accepts an input shape of (1,7), representing a vector of records from the past 7 days of revenue. An activation function of *ReLU* (Rectified Linear Unit) is utilized, defined as $f(x) = \max(0, x)$, which enables only zero or positive neurons to be activated. The second layer is a Dense layer with 7 units, also employing the ReLU activation function. Finally, we have a Dense layer comprising a single unit, responsible for predicting the output for the next day.

6.4.2 ***Model Training.*** The LSTM model was trained using the Adam optimizer with default hyperparameters, including a learning rate of 0.001, $\beta_1 = 0.09$, and $\beta_2 = 0.999$. Mean squared error (MSE) was utilized as the loss function during training. Training was conducted for 100 epochs with a batch size of 124. Additionally, a validation split of 0.1, representing 10% of the training data, was utilized. The best achieved Root Mean Square Error (RMSE) was 352.9962, used as the evaluation metric during training. The training history of the LSTM model is illustrated in Figure 17.

6.4.3 ***Testing and Forecasting.*** As before we apply the model to the testing data and forecast if for a further 100 that has a vector of records of the last 7 previous days. The RMSE (5.5) achieved on

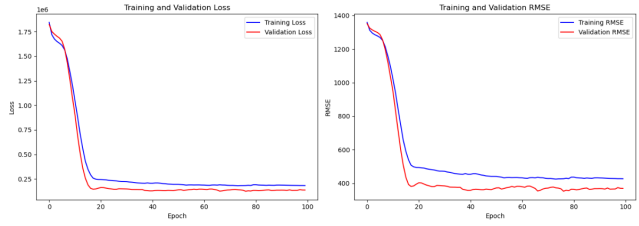

Figure 17: Training and Validation Metrics (LSTM)

the testing data was 327.863. The Figure18 illustrates the actual and predicted data using the LSTM model.

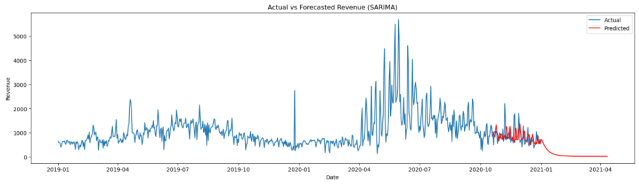

Figure 18: Actual, predicted and forecasted data (LSTM)

7 CONCLUSION

This study has presented an insightful exploration into the dynamics of bike-sharing demand prediction for Santander Cycles in London, leveraging sophisticated machine learning models. Among the various models tested, the Gradient Boosted Trees (GBT) model stood out for its precision in forecasting demand to its proficiency in managing complex, non-linear relationships inherent in the data. The comprehensive data analysis, encompassing weather conditions, public holidays, and proximity to public transportation, enabled the GBT model to predict total bike rentals and consequently revenue with a high degree of accuracy. The application of these predictive models not only supports operational decision-making but also provides a strategic framework for enhancing urban mobility services. By incorporating environmental and temporal variables, the models could capture seasonal trends and daily weather fluctuations, offering a nuanced understanding of urban transport dynamics.

Having quantified the total rental duration per day, projecting and forecasting revenue for the subsequent 100 days based on this duration would greatly aid Santander in managing and maintaining their cycles following the anticipated demand. Our analysis revealed that SARIMA achieved the lowest RMSE of 321.350 on the testing data, followed closely by the LSTM model with an RMSE of 327.863, and finally the ARIMA model with an RMSE of 403.691. As depicted in Figure 20, the ARIMA model predicts stable revenue, while both SARIMA and LSTM models, with higher testing data accuracy, forecast a decline in total revenue. However, the accuracy of these forecasts can only be validated over time. However, it's important to note that forecasting is inherently uncertain, and real-world factors may influence actual revenue outcomes differently than predicted. Therefore, continuous monitoring and refinement of forecasting models are essential for Santander to adapt effectively to changing demand dynamics and optimize resource allocation for their bike rental service.

8 LIMITATIONS AND FUTURE SCOPE

- **Larger grid search**: Currently, the LSTM model uses some base parameters. The model performance can be fine-tuned and improved further by conducting a grid search of parameters over a larger range of parameters.
- **Integrating Spatial Data**: Despite computational and time constraints that made spatial data integration challenging, leveraging distances from bus and tube stations to cycle stops could significantly refine model accuracy.
- **Segmented Models**: The current research did not employ model variations for training across different periods and stations. Future work could adopt these segmented models to potentially improve revenue predictions, with sufficient computational resources.
- **Advanced Deep Learning Techniques**: The application of deep learning, such as Recurrent Neural Networks, promises to reveal more complex patterns in data sequences, enhancing forecasting precision.
- **User Behavior Analysis**: Investigating user behavioral trends through clustering could lead to more personalized service and efficient bike distribution.
- **Environmental Impact Assessment**: Assessing the sustainability impact of bike-sharing schemes through predictive modeling is a critical area for future exploration, vital for urban transport planning.
- **Challenges with Model Scalability**: Attempted implementation of Lasso and Ridge regression and other demand prediction models with hyperparameter tuning and cross-validation faced computational barriers. The large dataset size restricted the ability to run these models effectively, preventing the generation of outputs.

Further research avenues could explore segmented modeling, a technique not extensively examined in this analysis. The models utilized in this study were predominantly configured with default parameters, except for ARIMA and SARIMA models. Additionally, the LSTM model employed was a basic Vanilla LSTM architecture. To advance our understanding and prediction capabilities, future work could introduce more sophisticated models and explore a broader range of deep learning architectures.

Specifically, incorporating advanced deep learning models such as Recurrent Neural Networks (RNNs), bidirectional LSTMs, and attention mechanisms could enable the models to capture more intricate temporal dependencies and patterns in the data. By leveraging these techniques, we can enhance the models' ability to learn from sequential data and improve forecasting accuracy. Moreover, exploring ensemble methods that combine predictions from multiple models could further enhance prediction performance and robustness.

REFERENCES

[1] [n. d.]. Link to our repository: https://github.com/lse-st446/project-2024-group-3/tree/ee41da9a3422a8b2731104195e724e1f926d3e37. ([n. d.]).
[2] Leo Breiman. 2001. Random forests. *Machine Learning* 45, 1 (2001), 5–32.

[3] Hardik Chhabra and Anubhav Chauhan. 2024. A Comparative Study of ARIMA and SARIMA Models to Forecast Lockdowns due to SARS-CoV-2. *Journal of Forecasting* (2024).

[4] Jamal Fattah, Latifa Ezzine, Zineb Aman, Haj Moussami, and Abdeslam Lachhab. 2018. Forecasting of demand using ARIMA model. *International Journal of Engineering Business Management* 10 (10 2018), 184797901880867. https://doi.org/10.1177/1847979018808673

[5] Jamal Fattah, Latifa Ezzine, Zineb Aman, and Haj El Moussami. 2018. Forecasting of demand using ARIMA model. *International Journal of Engineering Business Management* 10, 2 (October 2018), 184797901880867. https://doi.org/10.1177/1847979018808673 License: CC BY.

[6] Anubhav Chauhan Hardik Chhabra. 2023. A Comparative Study of ARIMA and SARIMA Models to Forecast Lockdowns due to SARS-CoV-2. (2023). https://doi.org/AdvTechBiolMed.11:399

[7] Md Isalm, Biswas, Dulal Haque, Md Hossain, and Md Shahzamal. 2023. An Effective Data Driven Approach to Predict Bike Rental Demand. (12 2023). https://doi.org/10.1109/STI59863.2023.10464738

[8] Tae Kim, Min Jae Park, Jiho Shin, and Sungwon Oh. 2022. Prediction of Bike Share Demand by Machine Learning: Role of Vehicle Accident as the New Feature. *International Journal of Business Analytics* 9 (01 2022), 1–16. https://doi.org/10.4018/IJBAN.288513

[9] Alexey Natekin and Alois Knoll. 2013. Gradient Boosting Machines, A Tutorial. *Frontiers in neurorobotics* 7 (12 2013), 21. https://doi.org/10.3389/fnbot.2013.00021

[10] Nikolaos Pandis. 2015. Comparison of 2 means (independent z test or independent t test). *American Journal of Orthodontics and Dentofacial Orthopedics* 148, 1 (2015), 191–192. https://doi.org/10.1016/j.ajodo.2015.05.012

[11] Xinyu Qian. 2024. Using machine learning for bike sharing demand prediction. *Theoretical and Natural Science* 30 (01 2024), 110–119. https://doi.org/10.54254/2753-8818/30/20241078

[12] Transport for London. [n. d.]. Santander Cycles: What you pay. https://tfl.gov.uk/modes/cycling/santander-cycles/what-you-pay [Accessed: April 22, 2024].

A APPENDIX

A.1 Additional Figures

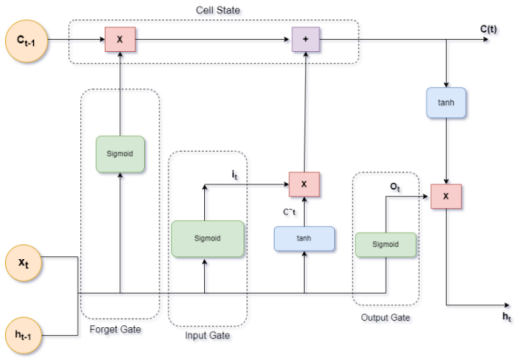

Figure 19: LSTM Architecture

	forecast_sarima	forecast_sarima	forecast_lstm
2020-10-19	1082.179890	1064.799291	1051.681519
2020-10-20	1007.556240	950.281767	898.790161
2020-10-21	1037.036864	922.749071	871.165833
2020-10-22	1125.378702	978.137052	748.762024
2020-10-23	1232.835623	1046.288826	1072.312256
...	...	...	...
2021-04-06	1083.717002	470.667830	42.229145
2021-04-07	1081.204483	468.437856	41.870224
2021-04-08	1084.439346	466.218448	41.521614
2021-04-09	1090.841976	464.009554	41.182892
2021-04-10	1095.589381	461.811127	40.853676

174 rows × 3 columns

Figure 20: Forecasts for all 3 models

```
[Stage 15:=============================================================>(105 + 1) / 106]
Chi-Square Statistic: 0.0
P-value: 0.0
There is a significant relationship between the 'start_station_name' and 'end_station_name' columns.
```

Figure 21: Chisquare Test

INDIVIDUAL CONTRIBUTIONS

Link to our group repository can be found here: [1].

(1) Candidate 26527: Contributed 25% towards the final project. This candidate was responsible for the revenue analysis and forecasting.
Files:
- Santander_Modelling.ipynb

(2) Candidate 24738: Contributed 25% towards the final project. This candidate was responsible for data pre-processing and integrating the final dataset for demand prediction.
Files:
- Exploratory Data Analysis.ipynb

(3) Candidate 25411: Contributed 25% towards the final project. This candidate was responsible for the geospatial and exploratory data analysis
Files:
- Exploratory Data Analysis.ipynb

(4) Candidate 34553: Contributed 25% towards the final project. This candidate was responsible for the demand prediction analysis.
Files:
- Santander_Modelling.ipynb