Report for Data Science Lab: Restaurant time series

University of Milano-Bicocca
Data Science master's degree
AY 2023-2024

Eleonora Brambatti - 858098
Costanza Pagnin - 898306
Marta Privitera - 898017

7^{th} February 2024

Abstract

This report describes the steps taken to carry out the Data Science Lab project. The project consists in analysing historical data concerning daily restaurant sales from a period of time spanning from the beginning of 2018 until the beginning of May 2023 for six establishments. The goal of the project is to estimate the optimal forecasting model for the rest of the 2023 year for each one of them. Since it is paramount that the data present trends in time and for them not to be influenced by anomalous events, a determining date from when to consider the starting point of the analysis was established. We first implemented simple ARIMA models, from the 2nd to the 6th order, followed by ARIMAX models with and without dummy variables. Finally, we resorted to Machine Learning models such as KNN, LSTM and GRU.

1 Introduction

The report is structured in the following way: at first the structure of the dataset will be described, followed by the relative preprocessing steps. This will include an explanation concerning the removal of certain data points due to their unusual characteristics. Subsequently, we will delve into the models applied to each restaurant. Finally, a conclusive analysis is conducted to establish which model was optimal for the restaurant. The steps which will be outlined from now on are shown only on the first restaurant but were conducted on each one of them.

2 Time Series Data

Exploratory Analysis

The original dataset (named 'restaurants') was made up by four columns: the daily receipt value (representing the number of receipts issued by the specific restaurant on that particular day), daily revenues, the given restaurant, and the given date. The time-frame spans from the 1st January 2018 until 3rd May 2023. The 'restaurants' dataset was split according to the restaurant where for each of them columns specifying the year, month and day of the week were added. When considering the variables available for analysis, we were curious to investigate the relationship between the receipt values and the revenue values. We proceeded by checking the correlation between the two variables, as they showed similar patterns. In order to visualise their relationship, a scatterplot was created (see fig. 1).

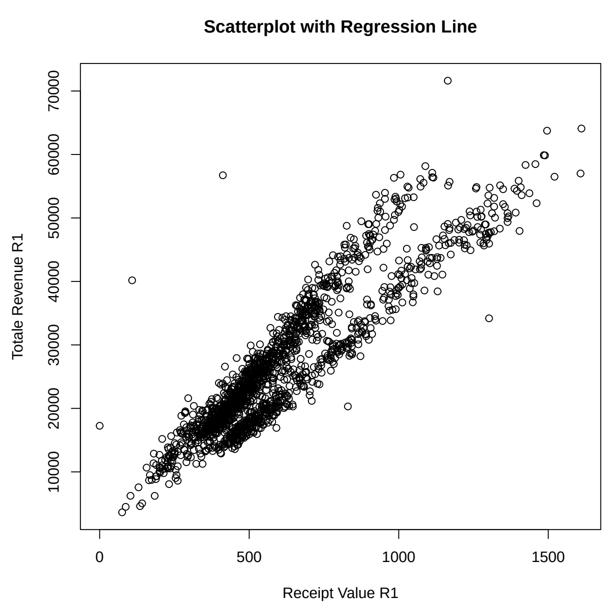

Figure 1: Revenue and receipt value Scatterplot for the first restaurant

In fact, it turns of they are very, if not perfectly, correlated. Their correlation coefficient value is 0.91. The result is unsurprising, as one would expect the value of the restaurant revenues to be linearly correlated to the value of the receipts it records.
We can observe that the two variables follow a linear trend, actually diverging in two lines. This behaviour may be due to factors such as the outbreak of the Covid pandemic, the different patterns observed between weekends and weekdays and so on. Given the two variables' similar behaviour, in the time series analysis we therefore decided to only consider as variables the value of the daily revenues together with their corresponding date.
To have an overview of the revenue values across restaurants, the density plot of revenue values for each one of them was depicted (fig. 2).

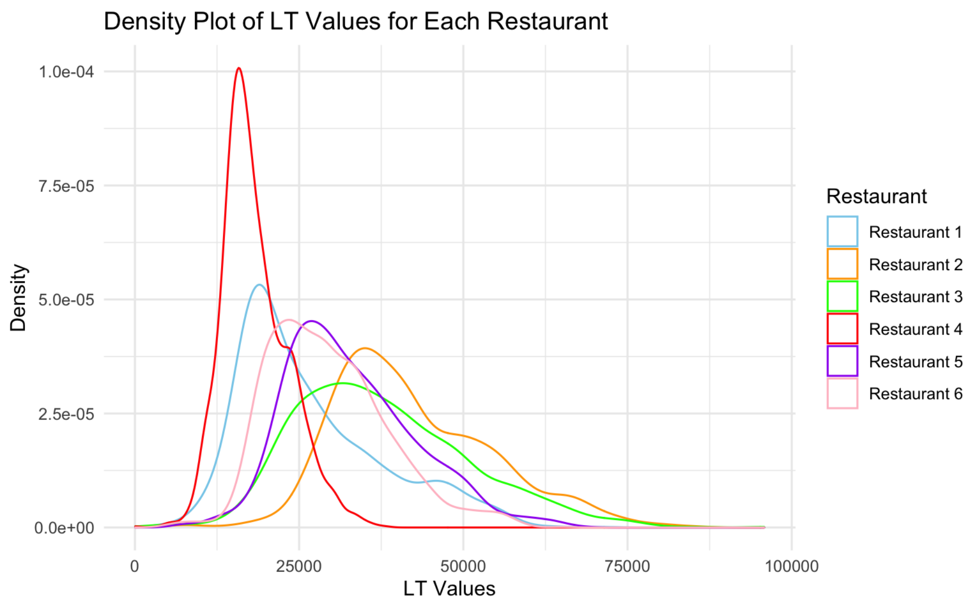

Figure 2: Restaurant density plots

We can observe that restaurant 4 exhibits lower variability compared to the others as well as a lower peak value. This means that this restaurant consistently hosts clients who tend to spend the same amount of money each meal. On the other hand, restaurants 2 and 3 in particular exhibit a higher variability and their peaks are centred on higher values, meaning that they cater for a wider economic range of clients, who have the possibility to spend both little or more significant amounts.
Finally, we created a boxplot (fig. 3) of daily revenue values for each day of the week to visually observe it there were differences in terms of expenditure.

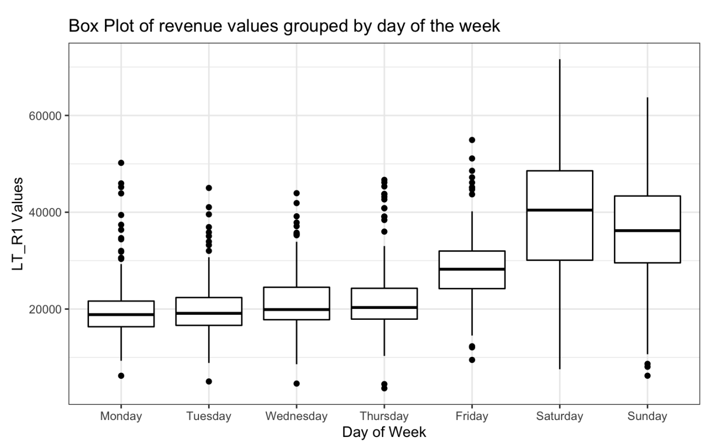

Figure 3: Boxplot of daily revenues by day of the week

It can be noted that from Monday to Friday there is little difference in spending patterns, both in terms of median expenditure value and outlier behaviour. Friday revenue start increasing, with the weekend resulting in the highest median and variability.

Missing values

Missing values [1] were an issue that had to be addressed. The first 8 months of 2018 were removed straight away since they only presented one value at the beginning of each month, probably because the data was being collected monthly up until then. Two other months with notable missing data were March and April 2020, coinciding with the onset of the first Covid outbreak. Specifically, for each restaurant, the following approaches were employed:

- **Restaurant 1:**
  - Removed the first 8 months of 2018.
  - Excluded March and April 2020.
  - Excluded May 2023.
- **Restaurant 2:**
  - Removed the first 8 months of 2018.
  - Excluded March and April 2020.
  - Excluded October 2021.
  - Excluded May 2023.
- **Restaurant 3:**
  - Removed all months of 2018.
  - Excluded all months up to October 2019.
  - Excluded March and April 2020.
  - Excluded May 2023.
- **Restaurant 4:**
  - Followed the same exclusion pattern as Restaurant 1 for corresponding years.
- **Restaurant 5:**
  - Followed the same exclusion pattern as Restaurant 1 for corresponding years.
- **Restaurant 6:**
  - Followed the same exclusion pattern as Restaurant 1 for corresponding years.

Other occasional data points were missing throughout the rest of the considered period and were completed by applying the Random Forest algorithm. In general, the number of missing data points per month remained in the range of 1-3, with some exceptions having 4, 6, or 7 missing data points in months influenced by the COVID-19 pandemic. The low frequency of monthly missing data points, particularly the prevalent presence of 1 or 2 missing data points, suggests that this is not strictly related to restaurant closures for rest or other reasons such as holidays. Usually, restaurants take advantage of such periods to increase revenue.
This observation, combined with the fact that Random Forest is trained considering data related to weekdays, guided the use of this machine learning technique for estimating missing data. Training

[1] It's important to note that in this context, missing data points actually refer to those rows where both the number of receipts and the total revenue are zero

Random Forest on weekly data is indicative as it takes into account weekly seasonality, considering the trend of total revenue and the number of receipts for prediction.
Another reason for using Random Forest is related to the length of the time series and the number of available data points. Filling in missing data points was considered beneficial to obtain a more complete and accurate dataset for predicting future data.
Regarding missing data that appears in only one of the two columns (either total revenue or receipts), we pursued the following reasoning:

- We preserved the non-missing data (either in the receipts column or in the total revenue column) if it was a value that made sense in relation to the other values in the distribution.
- Otherwise, we replaced it with the value determined by the Random Forest prediction.

Then, we predicted the other corresponding value in the column, the one that was missing, using the Random Forest algorithm. It's important to note that, in this case, the missing values are one, at most two, per restaurant.

Cut off point for beginning of analysis

In time series, it is important that historical data presents consistent trends in order for it to be reliable in forecasting future outcomes. However, the time span of the data includes the Covid 19 pandemic, which severely disrupted the hospitality industry. Hence, that data cannot be taken into account in the analysis. Our initial goal was to identify a point in time in post-covid years where the data was presenting similar trends to pre-covid times. Assuming that the behavior of people going to restaurants remains relatively consistent over time (those who go to eat choose more or less the same dishes) and in the same types of group, our aim was to identify a point in time in the post-Covid years where the data exhibited similar trends to pre-Covid times. In order to do so, we computed the percentage difference between the mean monthly number of the receipts between the last quarter of 2018 compared to the same months in 2019 (see table 1). The maximum percentage value was chosen as a benchmark to create tolerance limit.

Month	Mean 2018	Mean 2019	Percentage Difference	Tolerance	Tolerance Min 2019	Tolerance Max 2019
9	778	702	0.1	0.1	613	791
10	633	713	-0.1	0.1	623	804
11	699	769	-0.01	0.1	671	866
12	838	826	0.01	0.1	722	931

Table 1: Tolerance calculation

The tolerance limits of revenue values were calculated for every month starting from January 2021 (as we can see from the table in the next page: table 2).

Month	Min Tolerance	Max Tolerance	Mean 2021	Comparison 2021	Mean 2022	Comparison 2022	Mean 2023	Comparison 2023
1	613	790	334.4714	FALSE	521	FALSE	666	TRUE
2	623	804	425	FALSE	523	FALSE	644	TRUE
3	671	866	287	FALSE	504	FALSE	653	FALSE
4	722	931	403	FALSE	561	FALSE	729	TRUE
5	663	855	459	FALSE	555	FALSE	633	FALSE
6	629	811	507	FALSE	615	FALSE	NA	NA
7	701	904	581	FALSE	625	FALSE	NA	NA
8	659	850	518	FALSE	591	FALSE	NA	NA
9	613	791	541	FALSE	648	TRUE	NA	NA
10	623	804	554	FALSE	656	TRUE	NA	NA
11	671	866	547	FALSE	632	FALSE	NA	NA
12	722	931	599	FALSE	779	TRUE	NA	NA

Table 2: Monthly mean comparison

If the mean monthly value of receipts fell within the established tolerance limits or exceeded them consistently for consecutive months, it served as a reliable indicator for us to infer that the restaurant industry had successfully recovered, recording receipts amounts that were either similar to or even better than the pre-COVID levels. We immediately notice that the average number of receipts for all months of 2021 falls below the tolerance threshold. However, this phenomenon does not surprise us when considering the impacts of Covid-19 on the restaurant industry throughout the year 2021. It is important to note that the restaurants under analysis are located in the following areas: Montebello, Piacenza, Voghera, and Stradella. These areas were heavily affected by Covid-19-related restrictions. Indeed, the Lombardy and Emilia-Romagna regions faced particular challenges in the early months of 2021, especially with March and April characterized by the classification in the red/orange zone. This resulted in significant restrictions, including the prohibition of leaving home except for essential reasons and the consequent ban on dining in restaurants. This context justifies the observed decline in the data, and we are equally unsurprised by the slow recovery throughout 2022. In fact, until July of this year, the averages consistently remained below the tolerance level, only slightly returning within the acceptable range in the subsequent months of 2022, with the exceptions being November 2022, March, and May 2023, where the averages fell below the minimum tolerance level. Since we cannot regard 2021 as a post-COVID period, given the considerations made earlier regarding the regions' color classification and the introduced restrictions, we decide to establish January 2022 as the starting point for our time series analysis.

We can observe the section of the analysed time series timeframe in the figure below. The cut is performed with respect to the time series related to daily revenue.

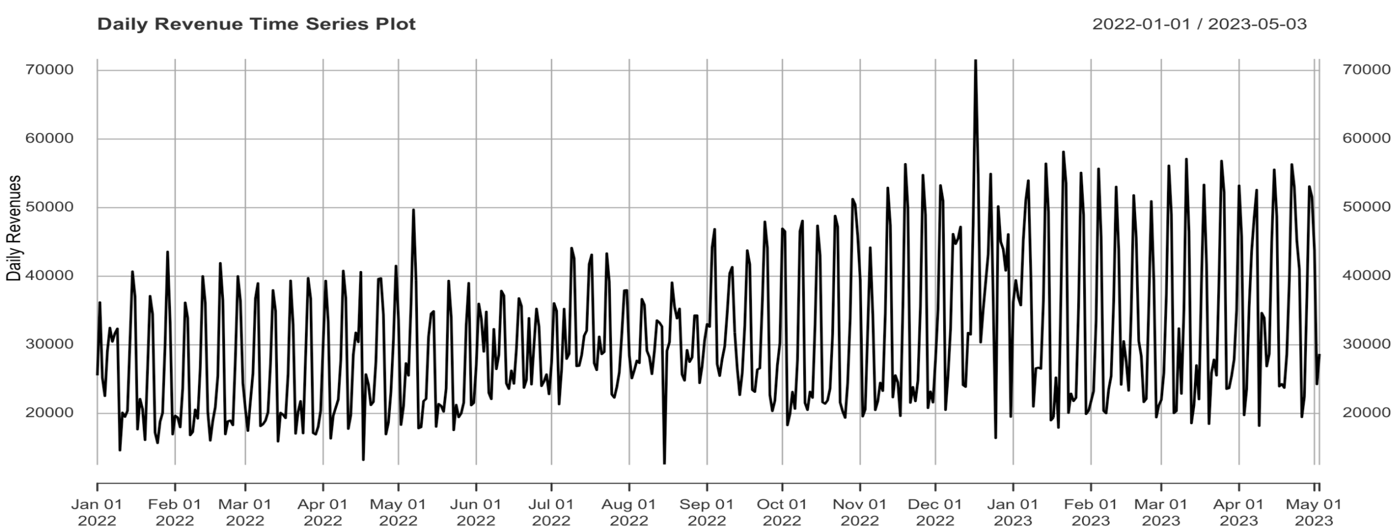

Figure 4: Time series Plot

3 Forecasting models

3.1 Arima

In time series analysis, the ARIMA (AutoRegressive Integrated Moving Average) model is a powerful tool for forecasting future values based on past observations. In our case, we started by performing rolling forecasting only on the day after the last date of the train set and iterating this procedure for all the month of April 2023, adding each time, the actual value of day previously predicted. The ARIMA model is characterized by three components: Autoregressive (AR), Integrated (I), and Moving Average (MA). The parameters (p, d, q) inputted in the code represent the orders of these components, respectively. We have applied different ARIMA models, which are outlined below, to find out which one fits our data best. In order to address the stationarity issue, to begin with, the logarithmic transformation was applied.

3.1.1 ARIMA Models

We started off with the ARIMA(2, 0, 0) forecasting Model. The single individual parts mean:

- AR(2): The model includes an autoregressive component of order 2, meaning that the current value depends linearly on the two previous values.
- Differencing (d=0): No differencing is applied, indicating that the time series is already stationary.
- MA(0): There is no moving average component in this initial model.
- The Seasonal Component (SARIMA) was added as well, which is introduces seasonal differencing (D=1) and seasonal orders (1,1,1).

We forecast the entire month of April considering one day per time. Our first train dataset was formed by the entire data of year 2022 and also the data relative to January, February and March of 2023. We first predict the value of total revenue for the 1^{st} of April and we calculated the MAE, MAPE and RMSE on the residual of this prediction and its actual value. We repeat this process for each day of april, saving each time the values of the three indexes. When we obtain the prediction for the last day of April we averaged all the indexes values in order to obtain a value representative for the entire month. In Figure 5, we can see the results obtained estimating an ARIMA model (2, 0, 0)(1, 1, 1), with frequency equal to 7 and drift. In the plot we can found the predictions for the month of April (in blue) compared to the actual values for that month (in green).

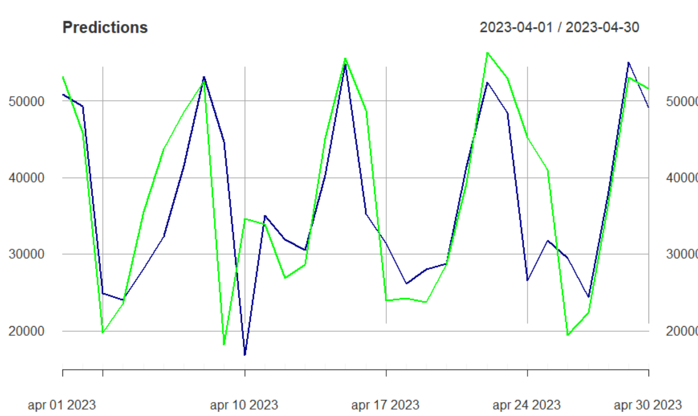

Figure 5: Total revenue predictions on the month of April (in blue) compared with the actual values of total revenue in the month of April (in green). The model estimated in order to obtain those prediction was an ARIMA model (2, 0, 0)(1, 1, 1) with drift and frequency equal to 7.

This model appears to have very high values of the RMSE, MAE, and MAPE which are respectively equal to 5.976861e+03, 7.335259e+07 and 1.942108e+01. These results can be explained by the fact that the total revenue variable assumes very high value (from figure 5 we can see that they are between 20000 and 50000). Considering this aspect, on one hand we can assume that these results are normal for this huge scale of values, but on the other hand, focusing on figure 5 we can see that the two paths of predictions and actual values are quite different, in fact they overlap only on few days like the $28^{th} and 29^{th}$. For these reasons that lead us to assume the non-optimality of the model we tried other models with a higher order of the AR component to see which model is associated to better results in terms of RMSE, MAPE and MAE. The autoregressive component order is increased incrementally from AR(3) to AR(6), introducing more historical dependencies into the model which enhance the model's ability to capture complex temporal relationships. In section 3.1.3 we can find all the results for each model.

3.1.2 ARIMAX Models

We then added a sinusoidal regressor to enhance each order of ARIMA model. The sinusoidal regressor captures recurring periodic patterns in the time series data, such as daily or weekly cycles. To facilitate model evaluation, the data was divided into training and testing sets. The ARIMAX model was trained on the dataset spanning from January 1, 2022, to March 31, 2023, incorporating both autoregressive and sinusoidal components. It's important to note that, in this scenario, we did not employ the rolling forecast technique. Instead, we used the data available from January 1, 2022, to March 31, 2023, as the training set. Subsequently, we made predictions on the data from April and the initial days of May 2023. This approach allowed us to assess the model's performance on the test set, generating errors that can serve as indicators for future forecasts on individual upcoming days. By observing the figure 6, one can see how well the ARIMAX2 model predicts future values by plotting the forecasted values alongside the actual values and providing a visual comparison (the plot represents our test set).

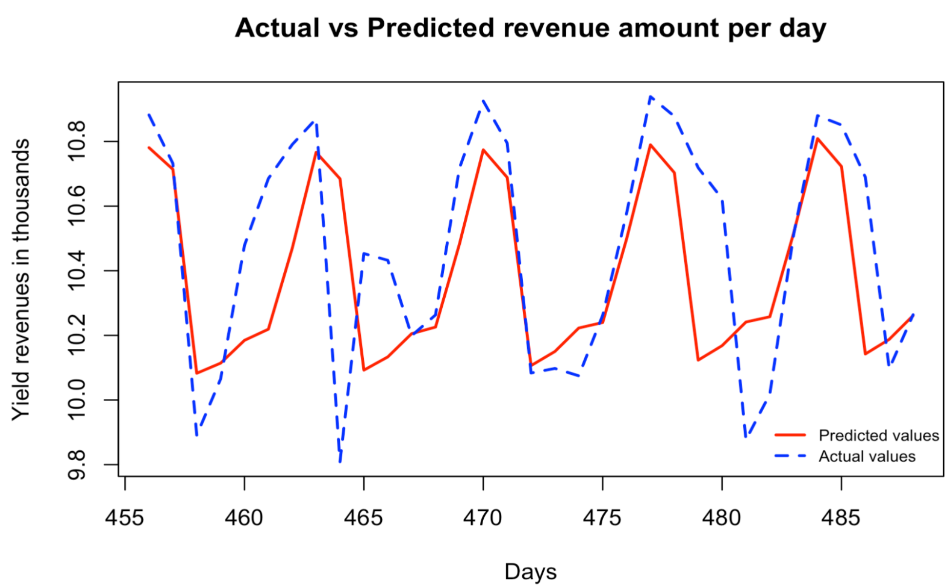

Figure 6: ARIMAX2 Predicted vs actual values. Each point on the x-axis corresponds to a day in 2023. The starting point, index 455, corresponds to April 1st, while the last point (488) corresponds to May 3rd, 2023. The curve in blue represents the test set on which predictions are made with the model.

We plotted the residuals of the model (calculated on the entire time series) in order to gain valuable insights into how effectively the model captures the underlying patterns in the data. Ideally, residuals should exhibit a random distribution centered around zero. Deviations from this pattern indicate areas where the model may face challenges. Specifically, identifying high residuals — those exceeding 0.05 or falling below -0.05 — pinpoint instances where the model struggles or encounters notable discrepancies in prediction accuracy (see fig. 7). These data points (corresponding to revenues per day) exhibiting high residual values were analysed and were actually points in time corresponding to or close to festivities. The high residual values are around 208 (Christmas day), 321 (Easter day), 322 (Easter Monday day), and near Ferragosto day.

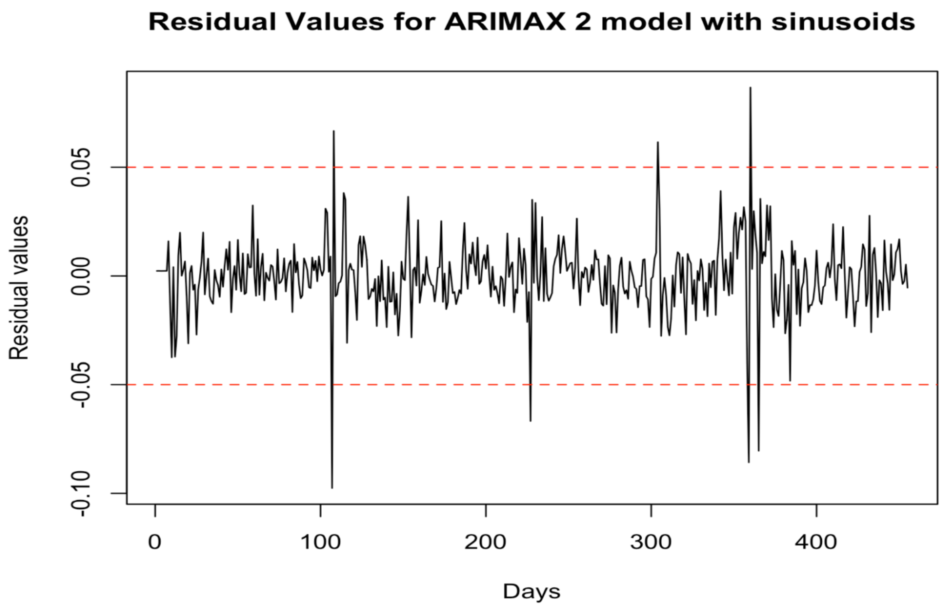

Figure 7: Residuals of ARIMAX model of order 2 with sinusoids. On the x-axis, each index corresponds to a day, spanning from the first day of January 2022 to the third day of May 2023.

We proceeded by modelling a new ARIMAX time series, using dummy variables for revenue values exhibiting high residual values exceeding the established threshold. The comparison between the predicted values and the actual values of the ARIMAX(2) dummy model is not presented in this context, as the values closely resemble those demonstrated for the test set predictions with sinusoidal variables. Displaying them would be redundant given the similarity, and thus, it is not meaningful to showcase both of them. The decision to omit the display of predicted values versus actual values for the ARIMAX(2) dummy model is further supported by the fact that the error metrics evaluated on the same test data are low and extremely similar for both models. The residual values for the entire time series follow the graph depicted in figure 8. We can see that the residuals span in a tighter value range than in the previous case, suggesting this model fits our data better. The ARIMAX model taking into account the dummy variables appears to yield more satisfactory results in terms of ME, RMSE, MAE, MPE, and MAPE compared to the model with just sinusoids (as we can see from next section where we compare metrics results). Therefore, the latter model is likely the better-performing model for our time series data between the two.

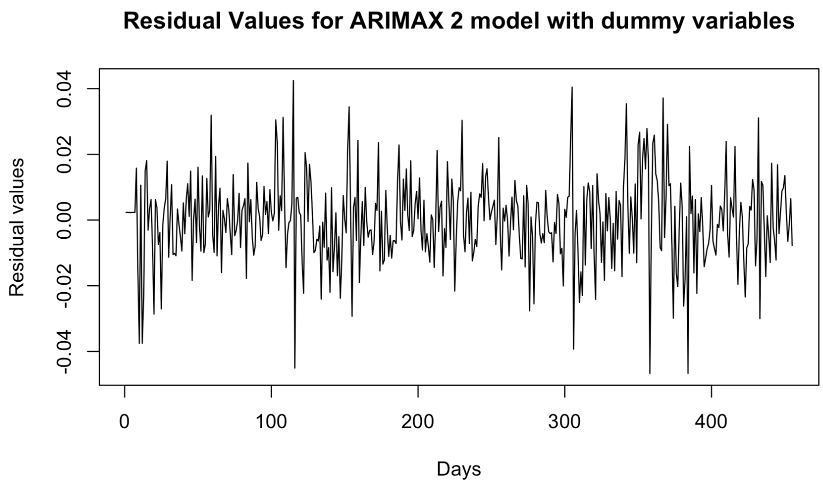

Figure 8: The plot illustrates the residuals associated with the adjusted ARIMAX2 model, incorporating dummy variables corresponding to the regressors that exhibited residuals outside the boundaries in the previous residual plot. In addition to the dummy variables, sinusoids are retained as regressors in this model. On the x-axis, each index corresponds to a day, spanning from the first day of January 2022 to the third day of May 2023.

The following image illustrates the forecasts made using the model with dummy variables, serving as an example of a forecast from April to May 2023 (test set).

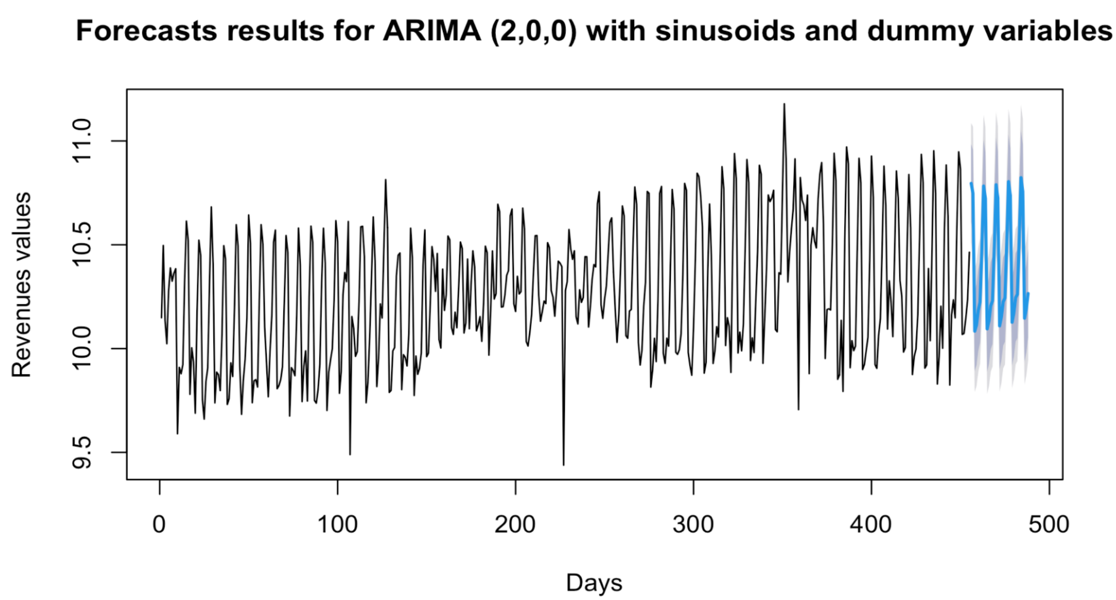

Figure 9: Time series ARIMAX2 dummy forecast. On the x-axis, each index corresponds to a day, spanning from the first day of January 2022 to the third day of May 2023. The blue line represents the forecasting of the revenues from April to May 2023, along with confidence intervals.

3.1.3 Comparison between Arima Models

The following tables illustrate the comparisons between Arima models of different AR order.
To determine which model is the best, we can consider the performance metrics and compare the values. For sure, lower values for RMSE, MAE, MPE, and MAPE are desirable. ME, which represents the bias in the predictions, should also be close to zero. The table below shows the metric values obtained on the test set associated to the rolling forecast with **ARIMA models**.

Model	RMSE	MAE	MAPE
ARIMA2	5.976861e+03	7.335259e+07	1.942108e+01
ARIMA3	5.944939e+03	7.305324e+07	1.933643e+01
ARIMA4	5.905268e+03	7.248408e+07	1.925557e+01
ARIMA5	5.922222e+03	7.269722e+07	1.929824e+01
ARIMA6	5.983918e+03	7.271404e+07	1.931547e+01

Table 3: Arima results comparison

Given these metrics, ARIMA4 might be considered slightly better. However, with such minor differences in RMSE and MAE, the models are very close in performance.
As far as the metrics related to the **ARIMAX models** are concerned here below we can observe the metrics related to different orders calculated on the test set.

Model	ME	RMSE	MAE	MPE	MAPE
ARIMA2	0.08098	0.28653	0.20493	0.713	1.9647
ARIMA3	0.08077	0.28640	0.20489	0.711	1.9643
ARIMA4	0.06515	0.28385	0.20012	0.562	1.9226
ARIMA5	0.0648	0.28383	0.20017	0.558	1.9232
ARIMA6	0.08077	0.28637	0.20353	0.712	1.9514

Table 4: Arimax result comparison

ARIMA4 and ARIMA5 appear to be the best-performing models among the ones compared. They both show similar lower error values across all metrics.
The results below are related to **ARIMAX models with dummy variables.**

Model	ME	RMSE	MAE	MPE	MAPE
Arima2	0.071	0.2874	0.2003	0.62	1.9231
Arima3	0.072	0.2876	0.2006	0.63	1.9257
Arima4	0.059	0.2861	0.1977	0.51	1.9011
Arima5	0.060	0.2864	0.1978	0.52	1.9021
Arima6	0.072	0.2881	0.2009	0.6345	1.9293

Table 5: Arimax dummy result comparison

ARIMA4 has the lowest RMSE, MAE, MPE, and MAPE values, indicating better overall performance in terms of accuracy and precision. ARIMA5 also performs well across all metrics. ARIMA3 and ARIMA2 have similar performance, but ARIMA2 has slightly lower values in some metrics.
Considering these metrics, ARIMA4 with dummy variables seems to be the best-performing model for our dataset.

3.2 Machine Learning models

We then proceeded to apply machine learning for time series forecasting. In this case, the data was split into train and test set, the latter composed by 30 days from 1^{st} of April 2023 until the 30^{th}. The models were evaluated on their goodness of fit on predictions compared to the test data.

3.2.1 KNN

We started off with the KNN algorithm, where two versions are implemented with different configurations, and their predictions are evaluated using various performance metrics.
More specifically future values are predicted based on the k Nearest Neighbors, i.e., the k most similar series to the last temporal lag preceding the values to be forecasted. In this case, a one-year lag of data was used as the temporal lag. Once the k most similar series were identified based on Euclidean distance, the 30 future values were predicted by averaging the 30 values that follow the identified k series. In performing this average, a preference was given to more recent series among the identified k series. The recursive method was chosen over the MIMO (Multi Input Multi Output) methodology to enable a one-step-ahead forecast. Through the recursive method, each iteration considers not only all the data in the series but also the forecasted data generated up to that iteration. This approach increases the sample base of the training set. In the following Figure we can observe the predicted values with respect to the actual values. On the x axis the Days are represented by the number of days which have passed from the 1st January 2022. It represents the month of April 2023. The same reasoning applies for the KNN2 model.

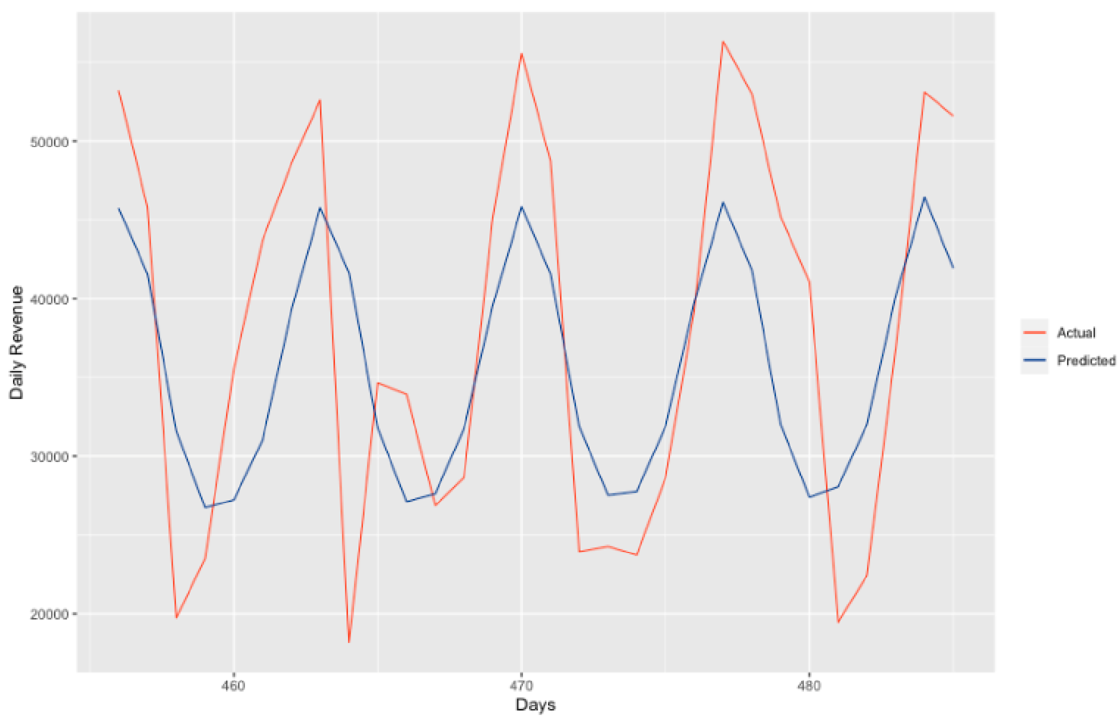

Figure 10: KNN1: Predicted vs actual values

Model	ME	RMSE	MAE	MPE	MAPE
KNN1	-2085.583	8915.614	7616.027	2.498916	0.234082

Table 6: KNN Model Evaluation Metrics for Model 1

The Mean Error represents the average difference between the predicted daily revenue values and the actual daily revenue values. A negative ME indicates that, on average, the predictions are underesti-

mating the actual values by approximately 2085.58 units.
The Root Mean Squared Error is a measure of the average magnitude of the errors. It is larger when there are larger errors. The RMSE of 8915.61 indicates that, on average, the predictions deviate from the actual values by approximately 8915.61 units.
The Mean Absolute Error is the average absolute difference between the predicted values and the actual values. The MAE of 7616.03 indicates that, on average, the absolute difference between predictions and actual values is approximately 7616.03 units.
The Mean Percentage Error represents the average percentage difference between the predicted values and the actual values. A positive MPE of 2.50 indicates that, on average, the predictions are overestimating the actual values by approximately 2.50%.
The Mean Absolute Percentage Error is the average absolute percentage difference between the predicted values and the actual values. The MAPE of 0.23 indicates that, on average, the absolute percentage difference between predictions and actual values is approximately 0.23%. Since we are dealing with daily revenue values which are on average around 30 thousand euros, the values exhibited above indicate moderate to high error levels.

Subsequently, a **second KNN version** was implemented, which involves using multiple KNN models with different values of k to generate predictions. The final forecast is obtained by averaging these values. Specifically, 9 KNN models were combined with k values of 30, 35, 40, 45, 50, 55, 60, 65, and 70. The forecasted values vs the predicted values are outlined below. The lines follow somewhat similar trends, however not similar enough to exhibit satisfactory metrics results.

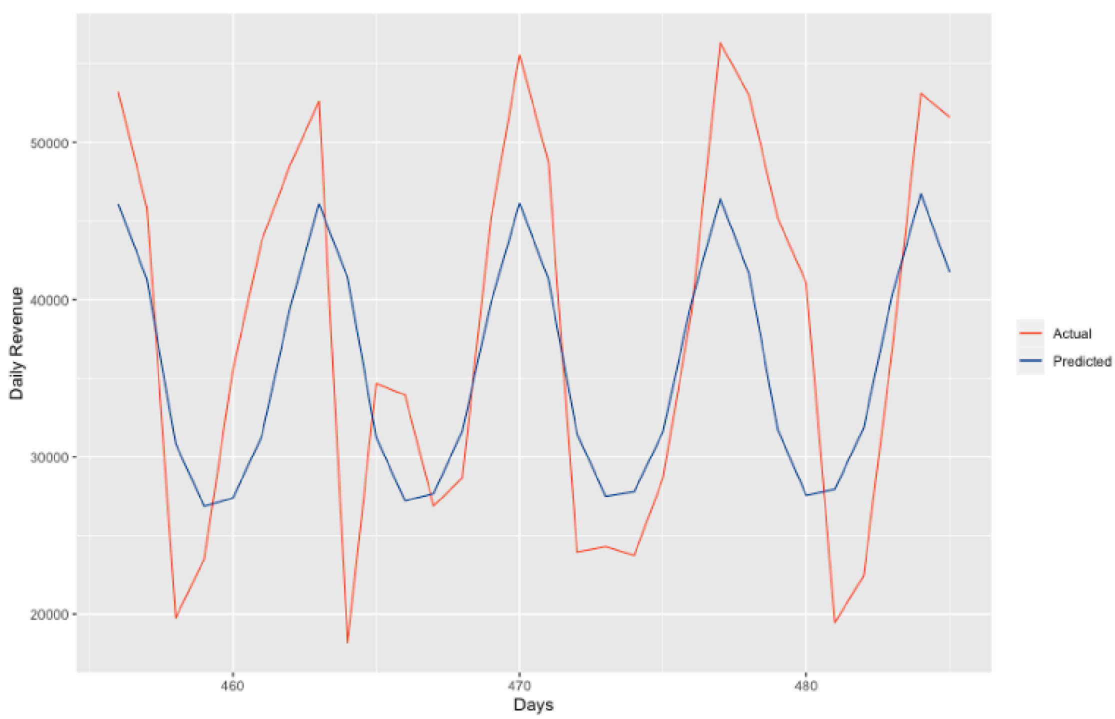

Figure 11: KNN2: Predicted vs actual values

Model	ME	RMSE	MAE	MPE	MAPE
KNN2	-2115.938	8795.65	7524.315	2.249234	0.2306353

Table 7: KNN Model Evaluation Metrics for Model 2 (KNN2)

The negative ME indicates that, on average, the model underestimates the actual values by approximately 2115.94 units. Similar to the interpretation for KNN1, the sign suggests a systematic bias, and

the magnitude should be considered relative to the scale of the data.
The RMSE of 8795.65 indicates the average magnitude of the errors. Given the context of the data, an RMSE of this size suggests a moderate level of error.
Mean Absolute Error (MAE): The MAE of 7524.315 indicates the average absolute difference between predictions and actual values.
The positive Mean Percentage Error of 2.25 indicates that, on average, the predictions are overestimating the actual values by approximately 2.25%
Mean Absolute Percentage Error (MAPE): The MAPE of 0.23 indicates the average absolute percentage difference between predictions and actual values.
In summary, KNN2 exhibits metric values slightly better than KNN1, thus we can conclude these models perform discreetly on our test data, considering the magnitude of the daily revenue values, which is on average 30 thousand euros.

3.2.2 RNN

We then proceeded by applying two recurrent neural network architectures: LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit). For each type of architecture, we implemented two versions, in order to improve the model fit on the data. The advantage of these types of algorithms is that they are able to memorise past information by sequentially analyzing the data. The data both in the LSTM and GRU case was subject to an initial pre-processing phase: they were initially scaled or, in the LSTM1 case specifically, were subject to a logarithmic transformation, and centered. Subsequently, they were arranged in the form of an array with 3 dimensions: the sample size, the number of lags, and the number of features. In this case, the time step is equal to 1, as well as the number of features.

In the **first LSTM model**, three LSTM layers are stacked, each capturing different aspects of temporal patterns. The first two LSTM layers return sequences, allowing the model to learn temporal dependencies over time, while the third layer returns a single output. Two dense layers are added, introducing non-linearities and enabling the model to capture complex relationships. The model is then compiled using the mean absolute error (MAE) as the loss function and the Adam optimizer with a customized learning rate. This model does not exhibit satisfactory results, probably due to the fact that it is too complex.

Due to the scarce results of the first one, a **second LSTM version** was implemented, yielding significantly better loss and forecasting results. This second version presents a single LSTM layer made up by 50 units, followed by a dense layer made up by 1 unit. It is therefore a more simplified version of the previous one. The model was then trained for 20 epochs, significantly fewer than in the previous version, with a batch size of 32. The MAPE value is 0.3543 and the figure depicting the real vs predicted values is shown below.

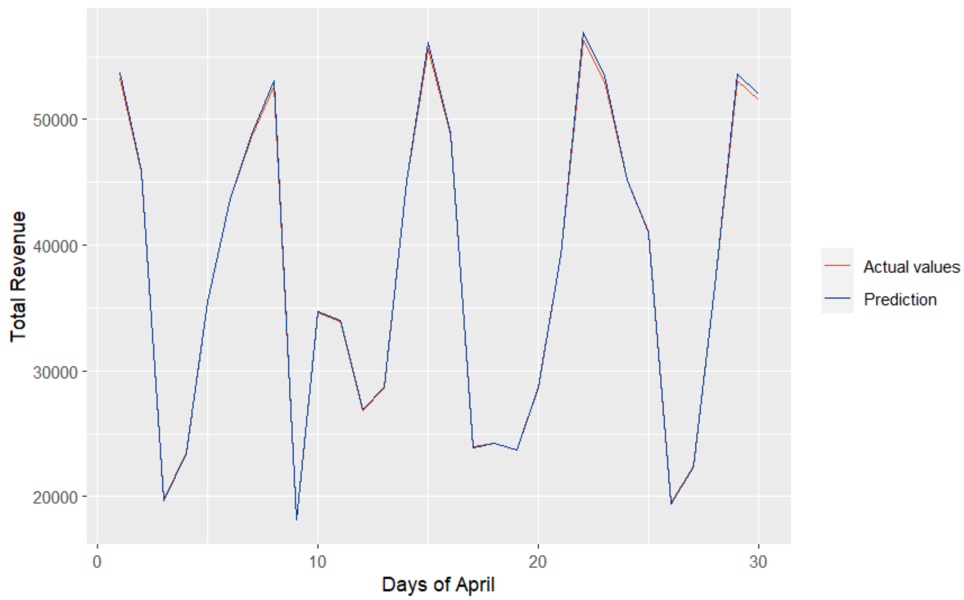

Figure 12: LSTM2: Predicted vs actual values of month April 2023.

From figure 12 we can see that the predictions are the same as the actual values with very minimal differences in contrast to the results obtained with the ARIMA models. We can consider the idea of being in overfitting.
In table 8 we can find all the values of the metrics ME, RMSE, MPE, MAE, MAPE

Model	ME	RMSE	MAE	MPE	MAPE
LSTM2	123.4385	247.2391	158.7407	0.2038467	0.3543625

Table 8: Evaluation metrics for RNN model LSTM2

The values of MPE (0.203) tells us that there is a tendency for the model to slightly overestimate the values on average. While the MAPE of 0.354 suggests that, on average, the absolute percentage difference between the predicted and actual values is 0.354%. The second type of model architecture (GRU) has a similar structure to the LSTM model architecture.

In the **first GRU version** we employed a simple GRU neural network for time series prediction using the Keras library in R. The model has one GRU layer with 90 units, followed by a dense output layer with a linear activation function. The mean absolute error is used as the loss function, and the Adam optimizer is employed for training. The results yielded by the model are very poor to say the least, both in terms of resulting metrics and in terms of predicted vs actual values.

We therefore implemented a **second version of the GRU model**. It consists of a single GRU layer with 50 units (neurons), followed by a dense output layer with 1 unit, which is suitable for predicting a single continuous value (typical for time series regression). The model is then compiled using the Adam optimizer and the mean squared error (MSE) is used as the loss function, which is appropriate for regression tasks. The model is trained for 20 epochs with a batch size of 32. The MAPE result is 0.6184 and the forecasted vs actual values are depicted below:

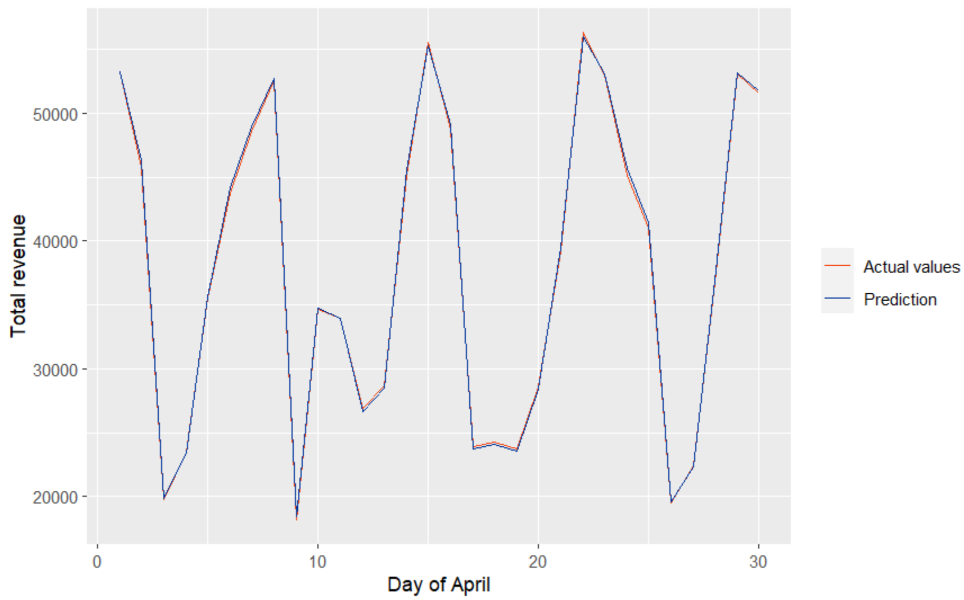

Figure 13: The plot depicts the forecast of total gross values on the test data, specifically for the month of April 2023. It is evident that the model appears to interpolate the data remarkably well, leading us to hypothesize that there might be an issue of overfitting.

In table 9 we can see all the values of metrics ME, RMSE, MAE, MPE

Model	ME	RMSE	MAE	MPE	MAPE
GRU2	93.07336	260.4498	222.3518	0.1980595	0.6184

Table 9: Evaluation metrics for RNN model GRU2

Comparing the results of table 9 with the ones in table 8 with LSTM2 indexes, we can see that here the MPE value (0.19) is smaller that the one estimated on the LSTM2 model suggesting a slightly better estimation of the model, on the contrary, the higher MAPE value of 0.6184 indicates a less reasonable (but still very good) level of accuracy in the model's predictions.

Judging from the tables of the Machine Learning Models and taking into account all the consideration done, KNN2 yields the best results.

3.3 Prophet model: rolling forecast

Prophet is a machine learning model developed by Facebook for time series forecasting. Specifically, Prophet is designed for forecasting time series data that exhibit seasonal patterns. It is particularly well-suited for business applications where the data may have multiple seasonal components. The model has become popular in the data science and business analytics communities for its ease of use and good performance on a wide range of forecasting tasks. We decided to use the Prophet model to compute a rolling forecast for one month on the daily revenue values. Firstly we split the dataset in train and test sets. The train part comprises data from the beginning of the year 2022 until March 31st 2023. We used the month of April 2023 in order to evaluate the goodness of the model. We developed the model on a rolling basis. This means that we trained the Prophet model on all the training data and we computed a one day forecast for April 1st 2023. We then added the estimated forecast value to the training dataset, from which the model was retrained to compute the following day forecast. This process was computed iteratively for the whole month of April.

We calculated these metrics in order to evaluate the goodness of the model

Model	ME	RMSE	MAE	MPE	MAPE
Prophet	1937.20	8751.80	7070.34	-2.23	22.31

Table 10: Evaluation Metrics for Prophet Model

E (Mean Error): The Mean Error represents the average forecast error. In this case, the average forecasted revenue is approximately 1937.20 units higher than the actual values. Since our data averages around 35,000, this error is relatively small in comparison.
The Root Mean Squared Error is a measure of the average magnitude of the forecast errors. It indicates that, on average, the forecasts deviate from the actual values by approximately 8751.80 units. Again, given that your data averages around 35,000, this suggests a moderate level of error.
The Mean Absolute Error is the average absolute difference between the forecasted and actual values. It indicates that, on average, the forecasts deviate from the actual values by approximately 7070.34 units. This suggests a moderate level of error.
The Mean Percentage Error expresses the average percentage difference between the forecasted and actual values. A negative value (-2.23) indicates that, on average, the forecasts are underestimating the actual values by approximately 2.23%. This suggests a slight tendency for the forecasts to be conservative.
The Mean Absolute Percentage Error is similar to MPE but based on absolute percentage differences. It indicates that, on average, the absolute percentage difference between the forecasts and actual values is approximately 22.31%. This suggests a moderate level of relative error in the forecasts.

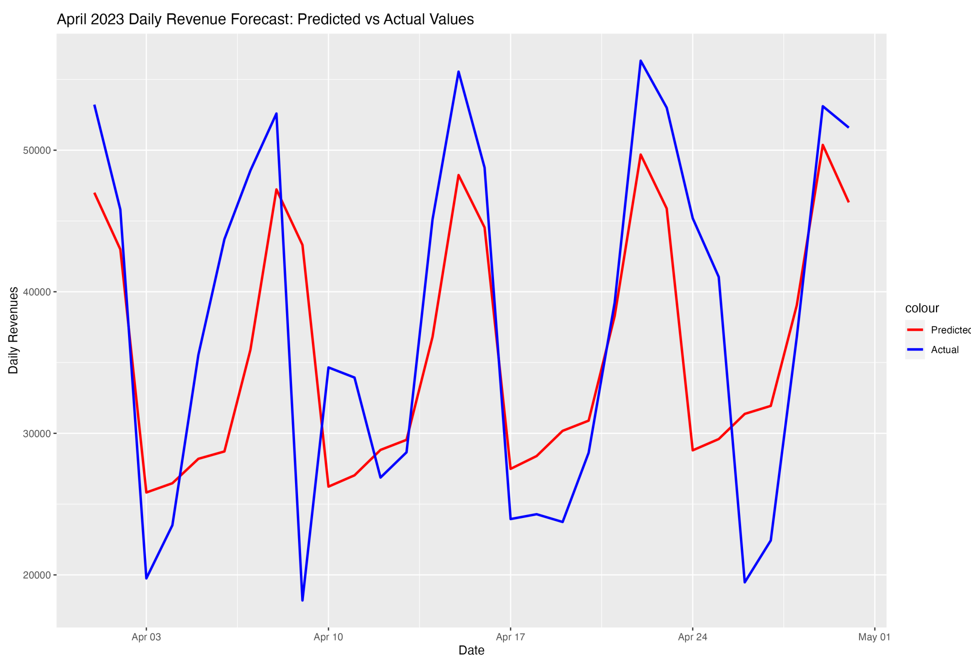

Figure 14: Prophet Rolling forecast: Predicted vs actual values

4 Final considerations and other restaurants

We tested several type of models and from the prevous sections we illustrated all the results we obtained in terms of ME, RMSE, MPE, MAE and MAPE.
If we want to indentify a best model overall we should choose the ARIMA models because the Machine Learning models are more prone to overfitting, while the ARIMA models are more appropriate to be applied on time series for forecast.

All the previous steps were taken to analyse the future daily revenue values for each of the six restaurants. We conducted descriptive analysis first, to then delve into the deployment of the ARIMA models, going from ARIMA2 to ARIMA6, as well as the ARIMAX and ARIMAX with dummy variables. Lastly, the same steps were taken for the Machine Learning models, where the KNN, LSTM and GRU models were employed. One restaurant that differs from others is Restaurant 3, where forecasting couldn't be performed due to the lack of data. As for the remainder of the restaurants, the models perform similarly to Restaurant1. The others yield values which can be observed in the R scripts.

5 Conclusion

We believe our work can be of great value to the restaurants we analysed. Having an idea of what future daily revenue values could occur gives restaurant owners a guideline in terms of how to manage stock orders, how to organise staff and also when to potentially renovate the restaurant, in order to make the closing time period match with periods of lower economic influx.
In addition to this, we can also consider the interesting aspect of our analysis that showed how, despite the fact that restaurants were one of the most affected sectors in Italy during and after the Covid period, the activities have resumed and have almost reached the pre-Covid level of number of clients per day, as a sign of no-stop both from people and food service sector.