Effects of Covid-19 on sales restaurants in northern Italy: a comparison for a quantitative assessment of losses

Alessandra D'Isabella: a.disabella1@campus.unimib.itl
Francesco Martinelli: f.martinelli21@campus.unimib.it
Cristiano Ruttico: c.ruttico@campus.unimib.it

ABSTRACT

Businesses like restaurants incur great variable costs for salaries, supplies and energy. When the economy flourishes the best set-up to maximize profit is to prefer fixed costs. On the other hand, variable costs have a huge advantage: their flexibility. That is the kind of cost you can work on when you are facing a shock like the Covid-19 pandemic. Said that you can reduce them, how much can they indeed restrain the loss of a period like the lockdown? We examined six catering businesses of northern Italy to answer this question by forecasting the sales during the first period of the covid pandemic. In this way it is possible to get a proxy of the impact of such an event on the 2021 revenues. To be able to predict, before we must find the best model to forecast on these businesses, that is why we implemented some of the most relevant models for forecasting on time series such as SARIMA and some innovative tools like PROPHET. In the analysis have been considered further relevant information, for example holidays, still with the goal of obtaining the best achievable prediction, to understand how much has indeed the first lockdown affected this kind of business.

KEYWORDS
Prophet; Forecast; Sales.

1. INTRODUCTION

A time series is a sequence of observations of one or more variables where the observations are temporally ordered. The time series is usually specified over a certain period and the points are recorded at regular intervals, such as on a monthly, weekly, or daily basis. The length of the time series required for a proper analysis depends mainly on the task desired. On time series different tasks can be performed, the main ones are forecasting, decomposition, impact analysis and clustering. We implemented forecasting from the beginning of the pandemic to understand the effects of this phenomenon. During the analysis, clustering has been also implemented, to find common patterns between the businesses and exploit the clusters. The case addressed the revenues of six businesses. The businesses analyzed in this case are restaurants located in Northern Italy, but no more information is given. For each restaurant We have data concerning the operational income, as sales and number of receipts. Due to the nature of the businesses, we can assume that the sales are essentially the whole revenues, so both terms can be used indiscriminately. The analysis has been conducted to discover patterns in the data useful to manage the businesses, as well: for example, discovering a particular seasonal pattern could be useful to implement a proper organization both in employees and storage, enabling the manager to optimize profit and cash flows. Investors are also interested parties in the revenue patterns prediction, so they can make informed decisions when it comes to expanding the business. The analysis has been conducted using python programming language and python libraries, mainly because of the flexibility and the vast number of tools that this instrument offers.

2. AIM OF THE ANALYSIS

The aim of the analysis is to develop a model useful to forecast the sales over time and to understand seasonal patterns in the data, moreover we wanted to comprehend the importance of the pandemic in terms of sales and revenues for this kind of business, in particular during the first lockdown. The assumption is that revenues are directly related to variable costs, for example if you know you will have twice the customers on weekends you could find a way to hire an employee just for weekends with the most profitable kind of contract. So, with this information the manager could properly operate optimizing the storage, the cash flows and the salaries relying on the notions provided by the model. For what concerns the Covid-19 Pandemic, it is complex to understand whether the shock set back the level of the series or it will be reabsorbed by a quicker growth on the longer period, so has been decided to focus on the short period computing the potential loss due to the first lockdown from the end of february till the end of may) and its effect on the yearly revenues and profit. In this way it should be possible to understand the importance of variable costs to contain the loss.

3. METHODOLOGICAL ASPECTS

To answer our research question, we exploited some state-of-the-art techniques to obtain the best possible results. From an overview and general perspective, we regarded the data as something to listen to; so, by means of a practical visualization tool we displayed the dataset at our disposal showing the behavior of sales during the reference period. We have also calculated some descriptive statistics using practical methods contained in python libraries; then, in order to forecast in the future, we exploited the functionality of some of the most common models to work with time series, in particular we considered the following models:

A) SARIMA and SARIMAX
B) PROPHET

Let's describe them briefly

A. SARIMAX and SARIMA

Understanding these models needs an introduction on what it is called the ARIMA model (AutoRegressive Integrated Moving Average): in the field of economics and statistics this model is a generalization of ARMA model (Autoregressive Moving Average): devoted to analyzing time series stationarity, i.e. a condition that guarantee that mean and variance are constant over time.

We will focus on the explanation of the ARIMA model, and then adding the seasonal component, we will explain SARIMA.

The ARIMA (p, dq) model is used for non-stationary time series without considering the seasonality; in the acronym of this model is encapsulated its functioning:

- **AR**: It is the 'AutoRegressive' term: this component is used to express the dependency between a variable and his lagged versions
- **I**: it is the 'Integrated' term: this component is the one related to differencing order(d) that has to be chosen to make the series stationary.
- **MA**: it is the 'Moving Average' term: this component encapsulates the relationship between an observation and its dependency in respect to a residual error through a moving average model applied to the lagged observations.

So, an ARIMA model is a stochastic process that can be reconducted to a non-stationary model (ARMA) by differencing it d-times. Using the backward operator B and polynomial form of AR and MA we can describe an ARIMA (1) process as:

$$\Phi(B)(1-B)^d Y_t = \Psi(B)\epsilon_t \qquad (1)$$

But, usually, real life phenomena are prone to seasonality, in this sense we need to extend our model in order to incorporate this effect to get the best modelling for our data structure; in this sense we obtain we need to utilize a model called SARIMA (p,d,q)×(P,D,Q) (*Eq. 2*), in which (p,d,q) are intended in the same as we described before, and (P,D,Q) are relative to seasonality.

$$\Phi(B)\Phi_s(B^s)(1-B^s)^D Y_t = \Psi(B)\Psi_s(B^s)\epsilon_t \qquad (2)$$

By adding the possibility of contemplate exogenous variables to SARIMA, we obtain the SARIMAX (Seasonal Autoregressive Integrated Moving Average Exogenous); this model requires the same parameters as before: (p,d,q)x(P,D,Q,s), but here we have also the parameter 's' which indicate the length of each season in terms of periods, in our case, days, before that the seasonal behavior can be spotted. This model is an expansion of the Sarima model that can consider different variables with the aim of improving forecast results. An example of this type of variable can be 'Holidays' that we tried to use to improve our sarima model to forecast the daily sales of each restaurant.

B. PROPHET

Prophet is a library provided by Facebook that represents both an easy to use and powerful tool for time series forecasting. The core model is based on additive models where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects, that is the main objective of our analysis. It works best with time series that have strong seasonal effects and several seasons of historical data and it is robust to missing data and shifts in the trend, and typically handles outliers well [6].
Prophet uses a decomposable time series model with three main components: trend, seasonality and holidays, combined according to the following equation:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t \qquad (3)$$

where g(t) is the trend function which models non-periodic changes in the value of the time series, s(t) represents periodic changes (e.g., weekly and yearly seasonality), and h(t) represents the effects of holidays which occur on potentially irregular schedules over one or more days. The error term represents any idiosyncratic changes which are not accommodated by the model.

This specification is similar to a generalized additive model [7], a class of regression models with potentially non-linear smoothers applied to the regressors.

By definition the growth function models the overall trend of the data. The novel idea behind Prophet is that growth trends can be present at all points in the data or can be altered at what Prophet calls "changepoints" [8], which are moments in the data that present a shift in direction. Prophet can detect changepoints automatically or allows the user to set them. It also allows to adjust the power the change points have in altering the growth function and the amount of data taken into account in automatic changepoint detection.

The growth function g(t) can assume 3 different behaviours:

- Linear growth: Prophet uses a set of piecewise linear equations with differing slopes between change points. When linear growth is used, the growth term will look similar to the classic y = mx + b from middle school, except the slope (m) and offset (b) are variable and will change value at each changepoint.
- Logistic growth: the growth term will look similar to a typical equation for a logistic curve (see equation 4), except that the carrying capacity (C) will vary as a function of time and the growth rate (k) and the offset(m) are variable and will change value at each change point.

$$g(t) = \frac{C(t)}{1+x^{-k(t-m)}} \qquad (4)$$

- Flat: Prophet allows to choose a flat trend when there is no growth over time (but there still may be seasonality). If set to flat the growth function will be a constant value.

The seasonality function s(t) is a Fourier Series as a function of time, which can approximate nearly any curve or in the case of Facebook Prophet, the seasonality (cyclical pattern) in the data:

$$s(t) = \sum_{n=1}^{N} \quad (a_n cos(\frac{2\pi nt}{P}) + b_n sin(\frac{2\pi nt}{P})) \qquad (5)$$

Prophet allows the choice of the Fourier order and between additive and multiplicative seasonality.

The holiday event function h(t) allows Facebook Prophet to adjust forecasting when a holiday or major event may change the forecast. It takes a list of dates (there are built-in dates of US holidays or ∑user defined dates) and when each date is present in the forecast adds or subtracts value from the forecast from the growth and seasonality terms based on historical data on the identified holiday dates.

This model brings quite a few advantages over a generative model like ARIMA:

- Flexibility: It can easily accommodate seasonality with multiple periods and let the user make different assumptions about trends.
- Unlike with ARIMA models, the measurements do not need to be regularly spaced, and we do not need to interpolate missing values e.g. from removing outliers.
- Fitting is very fast.
- The forecasting model has easily interpretable parameters that can be changed by the user to impose assumptions on the forecast.

4. DESCRIPTION OF DATA

The dataset has been provided by the University of Milan-Bicocca and It refers to the performance of six different businesses in the catering sector, before and during the Covid pandemic period. For each restaurant the given variables are the sales and the number of receipts over the period between the 1st of January 2017 and the 12th of April 2021 (the period still ends during Covid-19 pandemic), so 4 years and almost 4 months. The variables shown in the dataset are:

- Date: daily date (ordinal);
- Vendite1: amount of sales of Restaurant 1 (quantitative);
- Scontrini1: number of receipts of Restaurant 1 (quantitative);
- Vendite2: amount of sales of Restaurant 2 (quantitative);
- Scontrini2: number of receipts of Restaurant 2 (quantitative);
- Vendite3: amount of sales of Restaurant 3 (quantitative);
- Scontrini3: number of receipts of Restaurant 3 (quantitative);
- Vendite4: amount of sales of Restaurant 4 (quantitative);
- Scontrini4: number of receipts of Restaurant 4 (quantitative);

- Vendite5: amount of sales of Restaurant 5 (quantitative);
- Scontrini5: number of receipts of Restaurant 5 (quantitative);
- Vendite6: amount of sales of Restaurant 6 (quantitative);
- Scontrini6: number of receipts of Restaurant 6 (quantitative);

After a first data exploration it has been decided to consider only the sales variable for each restaurant (instead of both sales and receipts) due to the high correlation noticed between the sales and the number of receipts for each business.

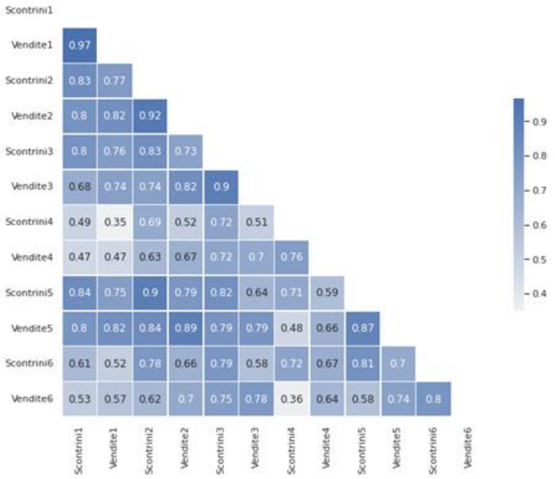


Figure 1. Correlation Matrix between quantitative variables

Figure 1 shows the Pearson correlation coefficient between the quantitative variables, computed through python library *pandas* method *.corr()*. The correlation is, in fact, around 0.87 on average, showing a remarkable level of correlation. So, the further analysis has been computed only on the sales variable to avoid the phenomenon of multicollinearity.

Other issues have been noticed, such as the high correlation between sales of different restaurants and a difference in the starting date for the time series of Restaurant 3 and 6 but these issues have been approached further.

5. DATA PRE-PROCESSING

The dataset presented three issues related to missing values. The first one was caused by two of the restaurants that have been open after the beginning of the time series: this means that the time series for these restaurants (3 and 6) start with a sequence of missing values, leaving only a short period before the exogenic shock (the pandemic). Another issue concerning missing values was related to how to approach the missing values due to sporadic business closure. For example, a tendency has been noticed for some of the restaurants to have missing values during Italian holidays. The last issue was represented by the forced closure of restaurants in Italy due to the pandemic from the 9th of March to the 16th of May 2020. This caused a consistent sequence of missing values.

The decision has been to approach the matter by interpolating the missing values with the average calculated between the previous and following values. To do so we used *scipy.interpolate* method *interp1d*. Another solution could have been filling the missing values with a constant value (0 could be a reasonable hypothesis because the revenues are indeed 0 on those days), but We decided that interpolation offered a better proxy for the performance, mainly for the sporadic missing values. The other two issues of missing values for a long period have been avoided by choosing for the computation only two of the time series. This choice has computational and smoothness reasons and will be more deeply explained in the next chapter.

Eventually, a new binary category has been created. Given the location and the nature of the businesses We decided to add a new variable, potentially explanatory. The new variable has been called *Holiday* and represents if that certain day is on a holiday period or not.

We used a proxy of the school holidays to represent the concept, because school holidays can quite properly represent phenomena like seasonal tourism. In particular, the considered periods were:

- From Christmas Eve to Epiphany
- Two weeks around Easter
- The period from the 10th of June to the 10th of September

6. ANALYSING DATA

DESCRIPTIVE ANALYSIS

To further explore the data, it has been decided to implement some descriptive analysis through python libraries for graphical representation, in particular using the libraries *seaborn* and *matplotlib.pyplot*.

The first step has been looking at the time series directly. In *Figure2* are shown the sales during 2017 of both Restaurant 4 and 5. The two time-series seem to vary around different levels, but peaks and valleys seem to manifest on the same days suggesting a dependence from an external variable. From the plot we can infer a certain degree of seasonality at least on a weekly basis.

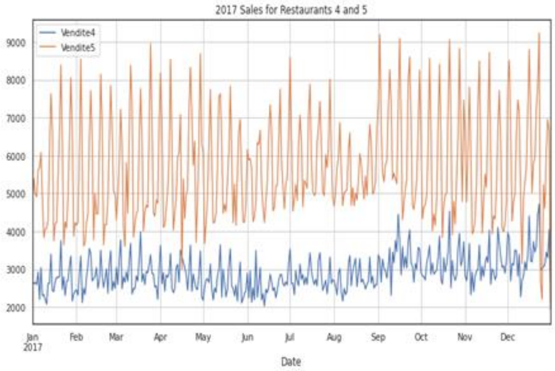


Figure 2. 2017 Sales for Restaurant 4 and 5 after the interpolation

We can then observe a series of boxplots representing the distribution of a random sample of observations grouped by different criteria. The boxplot gives a richer representation than average values because it considers the variation and the distribution of the variable. The boxplot consists of two whiskers representing the minimum and maximum value (outliers excluded), a box that represents the 50% of the ordinated distribution around the median (from the first quartile to the third quartile) and then the outliers. The outliers are represented by the diamonds and are computed as the observations far more than 1.5**InterquartileRange* from the nearer quartile, with the Interquartile Range calculated as:

$$IQR = 3^{rd}\ Quartile - 1^{st}\ Quartile$$

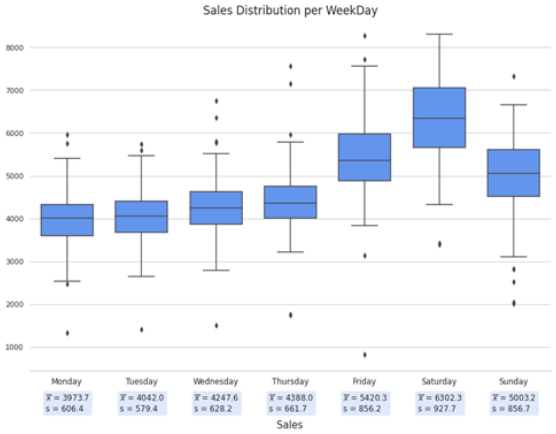


Figure 3. Boxplots for Sales distribution per WeekDay

The boxplot in *Figure3* represents the distribution of the observations grouped by day. The revenues are usually in the range between 2,000 and 8,000. The values show a strong week seasonality with consistently higher revenues during the weekend, on Saturday and Friday. With higher revenues also the variance of the observation tends to increase. All the distributions are consistently symmetric.

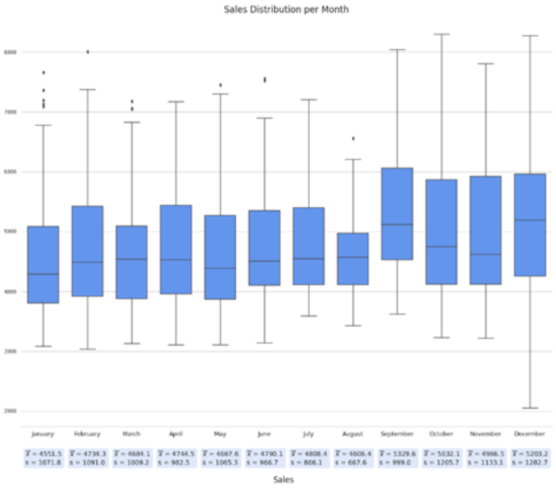


Figure 4. Boxplots for Sales distribution per Month

The second plot in *Figure 4* shows the distribution grouped by month. In this case the boxplots are complex to analyze because it does not seem to infer any remarkable observation: the most interesting feature is the high variance for the month of December, this could represent both the savings and the high expenses of the Christmas holiday period. Distributions per month are all positively asymmetric as a consequence of the daily distribution: noticing peaks during the weekends, the higher whisker is going to be constituted by weekend observations, but the density is higher around the median because there are more working days than weekend days.

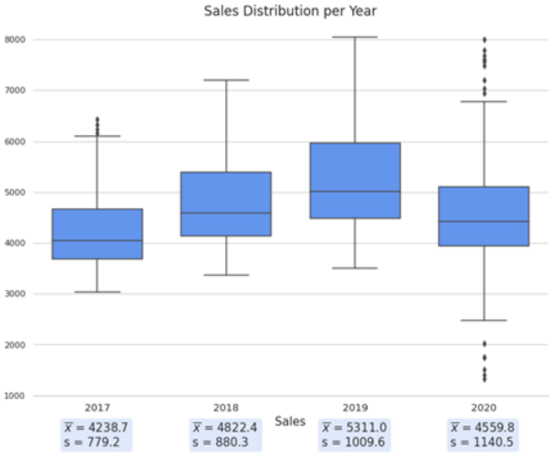


Figure 5. Boxplots for Sales distribution per Year

The third and last boxplot (*Figure 5*) is about years. It is evident the medium-term increasing trend and the arrest due to the pandemic. Has been decided to exclude the 2021 because data about that year were cut in April and the plot was potentially going to result biased by month seasonality. Here we can notice again the same positive asymmetry in the distributions, the causes are essentially the same as before. It is interesting to observe the high number of outliers in the year of 2020, they could represent the variation in the observations caused by the implementation of abrupt policies due to the pandemic.

In the previously shown correlation matrix (*Figure 1*) has been noticed a tendency of some of the restaurants to have a high Pearson correlation coefficient for sales. In *Figure 6*, the distribution of the sales observations have been plotted with the *seaborn.pairplot*. The pair plot is a useful tool to find patterns in the distribution of data and issues related to multicollinearity. The issue with multicollinearity between time series is the redundancy of the analysis on highly correlated series.

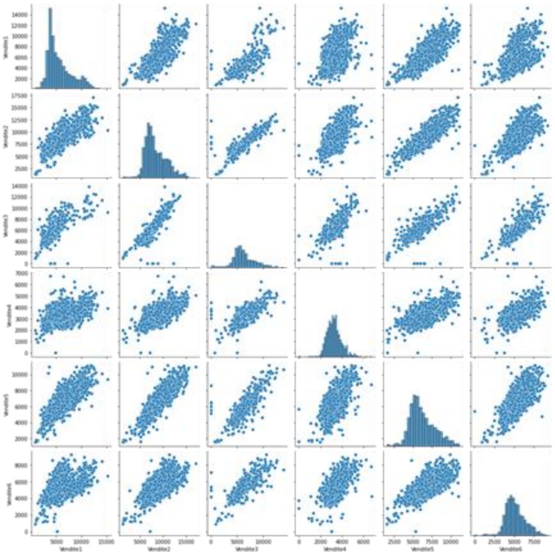
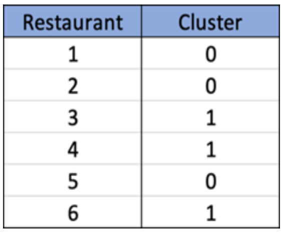


Figure 6. Pairplot for Sales

Figure 6 shows that the distribution of the data is located around the bisector for most of the time series. This means that we have further confirmation about the dependencies present in the data. One of the reasons can be identified in the seasonality patterns present in the observations that affect all the restaurants in a similar way, but this highlights the repetitiveness of a computational analysis implemented on all the six time-series.

CLUSTER ANALYSIS

The multicollinearity led to the decision of choosing only two series for the forecasting. The approach has been to exploit cluster analysis to choose representative series, looking for meaningful common patterns between them. For the analysis has been used python package *tslearn* and its function *TimeSeriesKMeans*; this package is built on other libraries as *numpy*, *scipy* and *scikit-learn*, a library that offers powerful tools for statistical and machine learning analysis. We decided to use the K-Means Algorithm to find a meaningful association between the series. The K-Means tries to find K not-overlapping subgroups (the so-called clusters) to split the observations in. K must be defined as an input variable; in this case we chose K=2 given the number of the time-series. The goal is to put in the same cluster the time-series more like each other and more different from the ones in other clusters. The concept of similarity is computed by calculating the *Euclidean distance* between the observations. The first step of the algorithm is to randomly select the centroids for the clusters, then It operates in an iterative way: the algorithm calculates the distance between observations and centroids, and assigns them to the nearer centroid cluster; the last step of the iteration is to calculate the mean of all data in the same cluster to compute the new centroid. The iteration is repeated until no change occurs, so the model converges [3]. The results are shown in the table below:



Restaurant	Cluster
1	0
2	0
3	1
4	1
5	0
6	1

To represent the clusters it has been decided to avoid the restaurants 3 and 6 because they were missing the first months of the time-series, so Restaurant 4 has been selected to represent cluster 1. To represent the second cluster the different choices were acceptable so Restaurant 5 has been selected.

PRELIMINARY ANALYSIS FOR MODELING

For the purposes of the research question, a subset was extracted from the available data; in particular, the period taken into consideration was from 22-02-2017 to 22-02-2020: i.e. an interval of exactly 3 years, with a total of 1096 observations.

Preliminary exploration and analysis of restaurant 4

In *Figure 7* below we show the data for the reference period for restaurant 4.

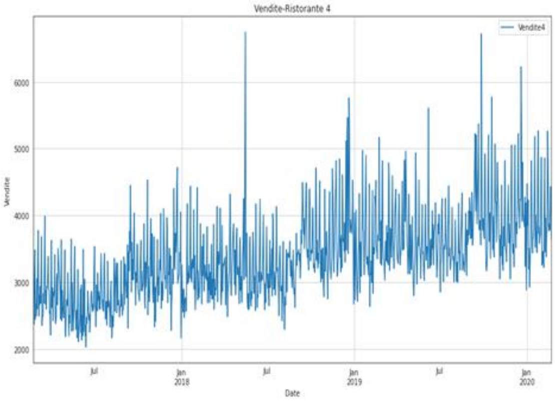


Figure 7. Sales Restaurant 4

On a first qualitative investigation of the data at our disposal, we note that there is a slight upward trend; we also note a certain repetitiveness in the peaks indicating a possible seasonality, which, analyzing the fact that the business in question is that of restaurants, we hypothesize could be weekly.

Intuitively then, our orientation would be to consider the series at our disposal as non-stationary. For a more rigorous, and specifically statistical, verification, we use the **augmented dickey fuller test**.

Stationary analysis restaurant 4

The dickey fuller test is one of the most widely used statistical tests, useful for determining the stationarity of a time series.

Applying the test, through the *adfuller* method of the *statsmodels* library, to our time series of restaurant 4 we obtain a p-value of 0.17 and a test statistic of -2.28: the values are summarized in the table at the end of the chapter (*Figure 13*).

With a significance of 95% and an observed p-value of 0.17 we can reject the null hypothesis and thus establish the non-stationarity of the series.

Seasonal analysis restaurant 4

For seasonality, we use the *seasonal_decompose* method provided by the *statsmodels* library.

Below, in Figure 8, we show the result of the decomposition of the time series for restaurant 4 with a temporal reference of one month; precisely, it goes from 02-12-2019 to 02-01-2020.

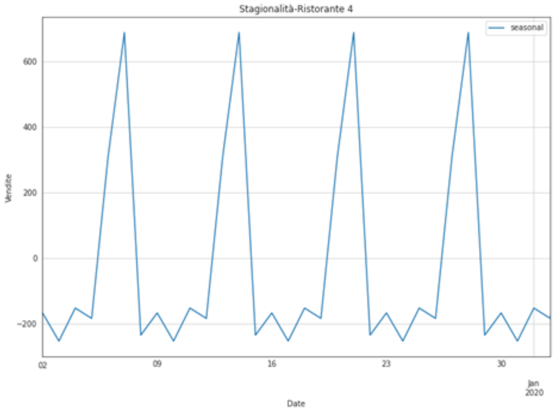


Figure 8. Seasonality Restaurant 4

As can be seen in the graph, we note that our assumptions about the seasonality of the time series were accurate as we are in the presence of a weekly seasonality.

Correlation analysis restaurant 4

Using the *plot_acf* method of the *statsmodels* library we can plot the autocorrelation graph for restaurant 4.

A serial autocorrelation/correlation plot is used to determine whether the time series data are positively, negatively, or independently correlated.

On the x-axis we have the lags of correlation evaluation, while on the y-axis we have the values assumed by our autocorrelation function.

For example, the correlation at the first lag is measured as the correlation of the time series values at time t with all the values at time t - 1 and so on.

We also have a confidence band that allows us to identify which correlations are significantly different from 0 and which are not.

Below in *Figure 9*, we report the autocorrelation analysis for restaurant 4

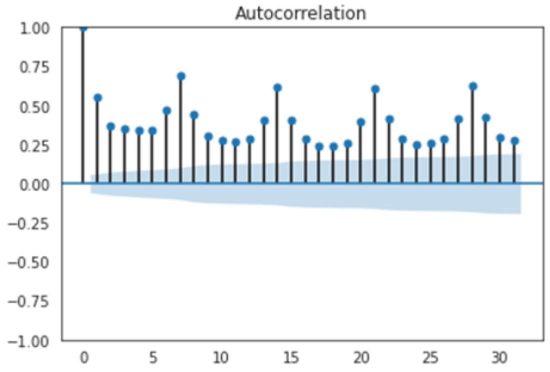


Figure 9. Autocorrelation Restaurant 4

We note a correlation above 0.5 at 7, 14, 21, and 28 lags, as we expected given the supposed weekly seasonality.

We also show the partial autocorrelation, which gives an idea of the correlation at different lags of the series, not considering the effects of intermediate lags.

In *Figure 10*, we can see the partial autocorrelation for restaurant 4.

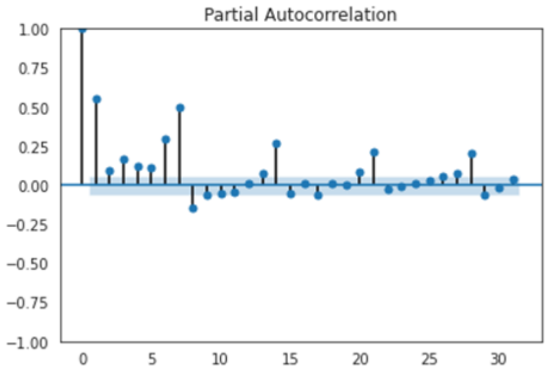


Figure 10. Partial Autocorrelation Restaurant 4

Preliminary exploration and analysis of restaurant 5

Figure 11 shows the data for the reference period for restaurant 5.

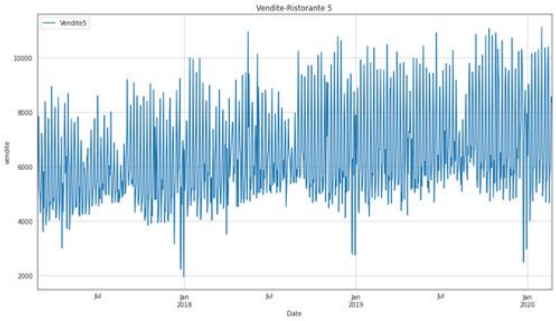


Figure 11. Sales Restaurant 5

In this case we can notice a slight increasing trend and we assume there could be a weekly seasonality in this restaurant as well, due to the type of business we are analyzing.

In this case too, we are going to carry out a stationarity analysis using the **dickey fuller test**.

Stationary analysis restaurant 5

Applying our test to the data from restaurant 5 we obtain a p-value of 0.10, and a test statistic of -2.53; again, considering a 95% confidence level and with an observed p-value of 0.10, we do not have enough confidence to reject the null hypothesis: the series will therefore be non-stationary.

The values are summarized in table (*Figure 13*).

Seasonal analysis restaurant 5

Similarly as we proceeded for restaurant 4, we use the *seasonal_decompose* method to access the seasonal component of the series. The result is shown below in *Figure 12*.

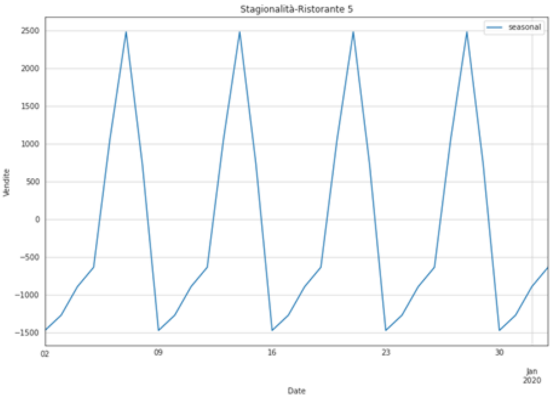
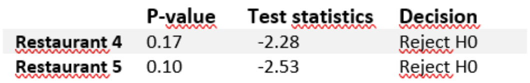


Figure 12. Seasonality Restaurant 5

We denote that the seasonality is weekly.

For restaurant 5 we do not report acf and pacf as the considerations in this respect are similar to those made for restaurant 4.



	P-value	Test statistics	Decision
Restaurant 4	0.17	-2.28	Reject H0
Restaurant 5	0.10	-2.53	Reject H0

Figure 13. Results of dickey fuller test for both restaurants

5. SARIMA

To obtain the orders of the sarima model, the *auto_arima* method was imported from the *pmdarima* library; the *auto_arima* method aims to identify the optimal parameters for the ARIMA model. The *auto_arima* method works by performing different tests to determine the differentiation parameter *d* and the parameter *p* for the AR (Auto-Regressive) part and the parameter *q* for the MA (Moving-Average) part. If the option concerning the seasonal component is enabled, auto-ARIMA also tries to identify the hyper-parameters: *P*, *Q* and *D*.

To find the best model, auto-ARIMA tries to optimize for a given information criterion: which can be 'aic', 'aicc', 'bic', 'hqic', 'oob which correspond to: Akaike Information Criterion, Corrected Akaike Information Criterion, Bayesian Information Criterion, HannanQuinn Information Criterion.

An information criterion is a method of comparing statistical models, used to evaluate which model provides the best parameters in order to balance the trade-off between the explanatory power of the model and the parsimony of the variables. In particular, the best model considering this trade-off is the one that provides the lowest value of the chosen information criterion [2].

5.1 RESTAURANT 4

Performing the *auto_arima* method on the data for restaurant 4 yielded the parameters: (4, 1, 0) x (1, 0, [1], 7).

Once the sarima model orders for restaurant 4 were obtained, the model was trained using the parameters obtained.

Once the trained model was obtained, it was then predicted on the test data for a period in order to evaluate the performance of the model.

Model performance:

For the purposes of assessing the model's performance, the **root mean square error** (RMSE) was used as a benchmark. This parameter is nothing more than the square root of the mean squared error in the forecasting phase. In formula we have:

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

The values obtained on the test set resulted in an RMSE of: 462.25 euros. Compared to the average of the test set of 3984.29 euros. We therefore noticed that the behavior of our model is quite good, as we have a lower RMSE than the average of the test set.

The following figure (*Figure 14)* shows the trend of the forecast obtained with the sarima model and the corresponding trend of the observed data.

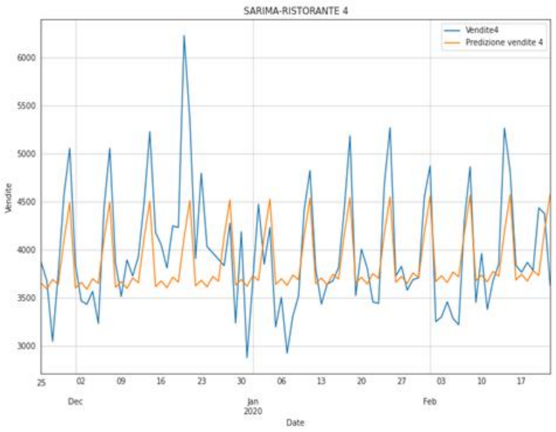


Figure 14. Restaurant 4 Model Evaluation

We notice that our model, in the case of Restaurant 4, performs quite well on the test set, intercepting seasonality, although never in a totally timely manner, especially in the period from 25 December 2019 to 5 January 2020.

5.2 RESTAURANT 5

Performing the *auto_arima* method on the data for restaurant 5 yielded the parameters:

(1, 1, 0)x(1, 0, [1], 7).

The obtained model was then fit to the training data and the prediction was performed on the test data.

Model performance:

The values obtained on the test set resulted in an RMSE of: 969.59 euros. Compared to the average of the test set of 7114.99 euros. We therefore note that the behaviour of our model is quite good as we have a lower RMSE than the average of the test set.

We therefore report below Figure *15*: in which we show the prediction on our test data and the trend of the observed data.

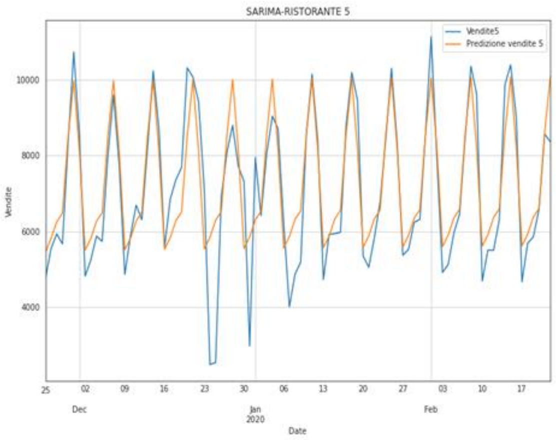


Figure 15. Restaurant 5 Model Evaluation

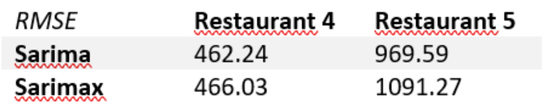
We can see that the model behaves slightly better in Restaurant 5, as it manages to capture the peaks due to the weekly seasonality on several occasions, and almost on time.

Especially in the period from around 9 January 2020 to 24 January 2020, the forecast follows the data at our disposal very well.

6. SARIMAX

To improve the performance of our models, we have added the option of taking additional variables into account to the *auto_arima* method.

In order to increase the discriminative power, in this case the variable chosen was 'Holiday'. This variable was created as previously described; unfortunately we obtained for both restaurant 4 and restaurant 5 slightly higher rmse values than those obtained with sarima. The values obtained specifically are shown in *Figure 16*:



RMSE	**Restaurant 4**	**Restaurant 5**
Sarima	462.24	969.59
Sarimax	466.03	1091.27

Figure 16. Comparing RMSE of different models for both restaurants

For this reason, predictions concerning the SARIMAX model have not been reported in the results.

7. PROPHET

7.1 RESTAURANT 4

As the last tool for forecasting we use Facebook Prophet. To follow what's been defined before, we use the same train-test split. In the following figures the blue lines indicate the trends, black dots indicate the real data, the light blue shadow indicates the confidence interval (uncertainty).
In figure 17 we can see the plot of the model forecast for the entire dataset (train + test) to give a glimpse both of how the model fitted train data and the test prediction (which we see without the black dots since we've hidden that data).

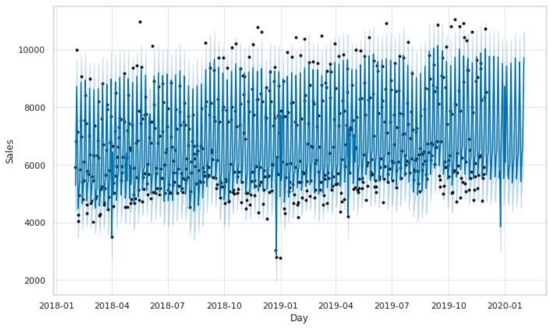


Fig. 17 Prophet’s prediction on the entire dataset

In figure 18 we can see each component of the model, i.e. trend, holidays, weekly seasonality, yearly seasonality and also monthly seasonality (which we had to add on top of the default components). It’s worth noticing that, for restaurant number 4, we can infer that it stays closed on holidays as it reaches negative peaks around -3000 euros. For what concerns weekly seasonalities we can observe that there’s a sales increase during the weekends, specifically on Friday and Saturday, while on Sundays we can infer that the restaurant’s closed. As per yearly seasonality we can observe that there’s a negative impact on summer while a rather positive trend in October and January. Monthly trends reach their peak in the central days of the month.

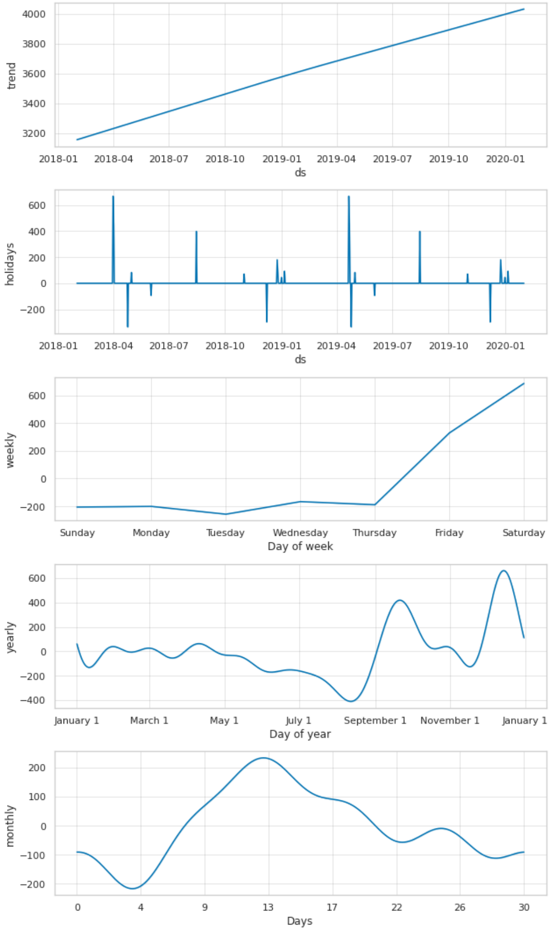


Fig. 18 Model’s seasonality components

We now investigate how the model performed on the testset and the forecast for the future (period of time that’s not present in the dataset). In figure 19 we can observe a comparison between the model’s prediction of the trends and the actual trends.

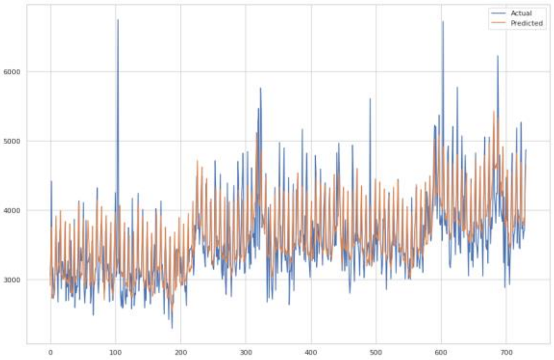


Fig. 19 Prophet’s prediction vs. actual trend

7.2 RESTAURANT 5

We now explore the results given by Prophet in the same way we did with restaurant 4. We can notice from the beginning that the variance in the data is much higher than restaurant 4 and so are the sales (figure 20).

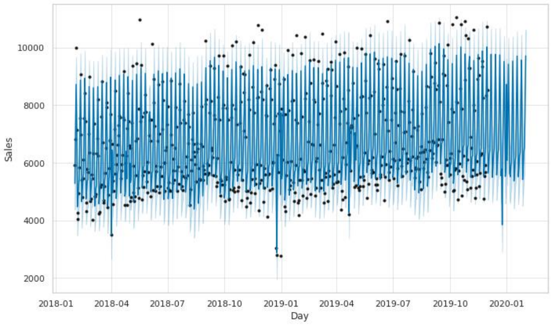


Fig. 20 Prophet’s prediction on the entire dataset

In figure 21 the model’s components are plotted. We can see that the holiday's seasonalities are rather different with respect to what we’ve seen for restaurant 4 as it shows some high positive peaks, suggesting that the restaurant is also operating during the holiday season. As for the weekly trend we can also notice a difference in the weekends as this restaurant is also open on Sundays as the positive trend suggests. Even the yearly seasonality presents a different behaviour suggesting that the restaurant is also open in summer. Lastly, the monthly seasonality also presents 3 peaks both in the central days and in the ending days of the month.

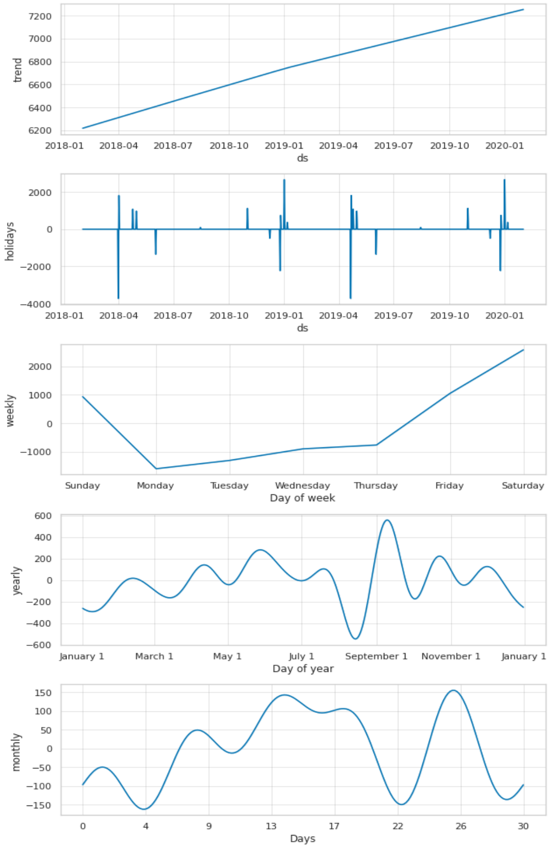


Fig. 21 Model’s seasonality components

We now take a look at the results given on the testset, which show a higher variance that is somehow captured by the model too (figure 22).

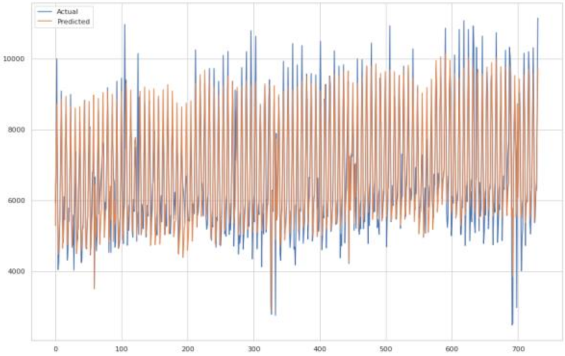


Fig. 22 Prophet’s prediction vs. actual trend

RESULTS

To follow the metrics defined earlier, we append the RMSE resulting from Prophet to the previously obtained ones from Sarima and Sarimax:

Table 1. Models Results

Models	**RMSE Rest. 4**	**RMSE Rest. 5**
Sarima	462.24	969.59
Sarimax	466.03	1091.27
Prophet	374.28	774.78

As we can see, Prophet performed better than both Sarima and Sarimax models on both clusters. We can also see that all models performed significantly better on Restaurant 4 and this is probably because Restaurant 4 is more regular in terms of seasonality whereas Restaurant 5 has some more isolated phenomena.

Getting more in detail we can notice that Restaurant 4 operates also during holidays, which means that data is provided for those days that are specifically modeled inside Prophet. As for Restaurant 5, which is almost always closed on holidays and on the weekends, the error can be caused by the interpolated data approximation.

Below, in figure 24 and 25, are shown the forecasts for both the series with Prophet.

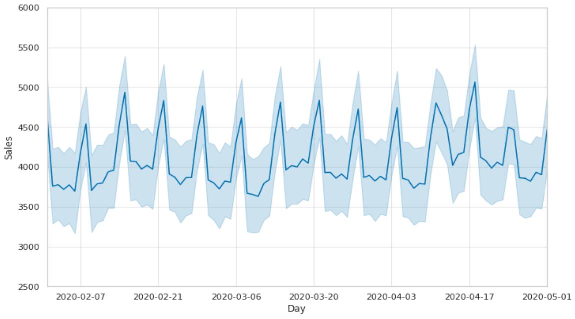


Fig. 23 Prophet's forecast for 3 months Restaurant 4

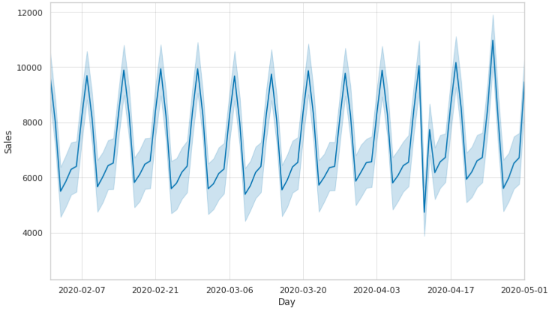


Fig. 23 Prophet's forecast for 3 months Restaurant 5

The forecasted period allowed us to estimate the potential revenues for the first lockdown period, by computing the sum of the predicted observations. In fact, since the businesses are closed the potential revenues can be interpreted as the estimated loss for the revenues due to that period. Dividing the estimated loss by the 2019 revenues (as a proxy for the revenues without Covid effects for 2020) has been calculated the impact of the lockdown on the businesses in terms of revenues.

Table 2. Estimated Loss for 2020

Rest.	**Est. Loss**	**2019 Rev**	**Est. Loss (%)**
4	373,116	1,373,434	27.17
5	663,656	2,547,719	26.05

CONCLUSION AND FURTHER DEVELOPMENTS

A strong weekly seasonality was noted for both time series. This was evident both from the elaborations performed with Prophet and those performed with Sarima. Prophet, on the other hand, has highlighted other monthly and annual patterns; if for the monthly ones there is no common interpretation between the clusters (it can be seen that restaurant 4 tends to have a peak in the middle of the month while restaurant 5 tends to have high values in the middle and at the end of the month), as far as the annual seasonality is concerned, it can be seen that there is a drop in the summer period, August in particular, and a peak in the September period: this confirms the suspicion that these activities are located in urbanized areas and not purely touristic as much as working, since there seem to be drops in correspondence to the periods in which people in Italy usually go on vacation.

The estimated loss due to the first lockdown in terms of revenues is around 26% for Cluster 0 and 27% for Cluster 1, so the impact has been similar between the two clusters. This result represents an upper value for the estimate loss on potential profit during the lockdown, in fact the estimate loss on potential profit could be easily calculated by subtracting the positive effect caused by variable costs saving. To compute this estimate, it would be sufficient to have the percentage of variable and fixed costs in relation to the period covered. With this amount the owners of the business can concretely estimate the effect of the lockdown on the annual profit.

Should one wish to analyze a longer forecasting period in order to analyze not only the effects of the first lockdown, but of Covid in general, a less granular analysis of the data is indicated that ignores short-term patterns and focuses rather on medium- to long-term trends. In this situation, a short-term analysis was deemed more appropriate, both because the available data did not lend themselves to a long-term forecasting horizon and because the effects of Covid are not (at the time of writing and for the available data) fully overcome.

Further analysis could be developed by integrating the data with more variables concerning, for example, the different kinds of costs: operational costs, ancillary costs, financial costs etc.. With this further information it could be possible to analyze the impact of each cost on the profit and perform a detailed analysis concerning the choices to adopt in order to properly manage the singular expense area.

REFERENCES

[1] Fattore M. (2020).Fundamentals of time series analysis, for the working data scientist (DRAFT).
[2]https://alkaline-ml.com/pmdarima/modules/generated/pmdarima.arima.auto_arima.html (20170, [Accessed: 21-Nov-2021].
[3]https://towardsdatascience.com/k-means-clustering-explained-4528df86a120 (2020), [Accessed: 23-Nov-2021].
[4]https://towardsdatascience.com/how-to-apply-k-means-clustering-to-time-series-data-28d04a8f7da3 (2020), [Accessed: 23-Nov-2021].

[5]https://www.istat.it/it/files/2021/04/Report_viaggiVacanze_2020.pdf (2021), [Accessed: 12-Dec-2021].

[6] DTaylor SJ, Letham B. 2017. Forecasting at scale. PeerJ Preprints 5:e3190v2 Working Paper version is available at https://doi.org/10.7287/peerj.preprints.3190v2. [Accessed: 16-Nov-2021].

[7] Hastie, T. & Tibshirani, R. (1987), Generalized additive models:some applications, Journal of the American Statistical Association82 (398), 371-386

[8]https://towardsdatascience.com/time-series-analysis-with-facebook-prophet-how-it-works-and-how-to-use-it-f15ecf2c0e3a, [Accessed: 03-Jan-2022].