LUISS

Master of Science in Data Science and Management

Course: Data Science in Action

Deloitte.

Business case: Evaluating the impact of an advertising campaign

Cantelmo Carlotta

Lona Masriera Marian

Martinese Marco

Introduction

This report refers to the study conducted on the business case: "Evaluating the impact of an advertising campaign". Our direct attention was into analysing the impact of an advertising campaign on the sales of a company that sells perfumes online. So the main goal was to verify if the ad campaign was effective or not. The advertising campaign started on December 3rd, 2019 and ended on December 24th of the same year.

Particular attention was paid to the seasonality of the time series and the difference between correlation and causation.
The ideal way to estimate advertising effectiveness is to get an estimate of the counterfactual: what would have happened without ad exposures. More generally, there is likely to be confounding factors that affect both exposure to ads and website visits (and the propensity of purchase), which make causal interpretations of observed relationships problematic.
Furthermore, if data at hand were in fact generated by ad managers who increase ad expenditure in periods of the year when the purchase probability is higher, then the observed relationship between ad campaigns and sales could be highly predictive for the historical data but not useful in predicting the causal impact of explicitly assigning additional resources to an ad campaign.

In the following paragraphs we explain how we overcame these problems.

Data

The used dataset, which was in the form of a csv file, was imported and combined into one.
We generated a new column called n sales to indicate the number of daily sales throughout the data preparation process. Obtained by dividing visits with convrate (conversion rate).
In order to conduct the following study, we constructed three additional datasets from the dataset created above for each city.

Methodology and modelling approach

As first thing we measured the correlation between visits and number of sales in the years of interest to see in which period the two variables are more correlated.

On the data sets created during the data cleaning phase we decided to apply a Multiple Linear Regression to predict the sales values in each city for each year of interest. The predictors used are visits and convrate. Results where plotted in order to better visualise relationships.
The decision to apply a Multiple Linear Regression was guided by the scope of comparing the actual sales values with those predicted and highlight differences among predicted and effective values. It is important to emphasise that the predicted values obtained through the Multiple Linear Regression don't take into account the effect of the campaign, while the actual values implicitly carry the effects of the campaign.

For the same purpose, we performed the same steps using the Decision Tree model in order to compare the accuracy of both models.

Then to check whether whether if sales depend on the number of visits we used the granger causality model.

Since the models used above are applicable to any type of data due to their generality we decided to use (for all the three given cities) models which are specific to time series in order to analyse the difference between predicted and actual values.
The models in question are:

- Arima
- Sarimax
- Prophet

In Arima, we used as variables firstly visits and then sales. Since from our measurements (carried out with the Dickey Fuller model and Rolling Statistics) the variable “visits” resulted to be non stationary with seasonality we transformed it using the logarithm into a stationary variable and eliminated seasonality. Rolling Statistics was recalculated in order to verify if the aim was achieved. The training set for this model comprises data from November 2018 to October 2019. (the month before the campaign began). The data for the test set was computed between November 2019 and February 2020. This is done so that the model can forecast the values of the test set without accounting for the campaign's impact.
The three order parameter values needed to match the Arima model were then discovered through the autocorrelation plot. The model fit was then done. We restored the original scale to the results (since they were previously logarithmically transformed).
Finally, we used the error measure to visualise the model (RMSE).

Sarimax is the second model that was used. This model can only be used for the variable visits because it is a non-stationary and seasonal model. The model's structure is identical to Arima's, with the exception of one new parameter: seasonal order.

Prophet, the last model utilised, does not require the data to be divided into train and test sets. Prophet requires the creation of a separate dataset because the model only reads two variables: ds and y. In ds it was placed the df index, which corresponds to the variable day, and the y corresponds to the visits. The model's fit was then determined, and we were able to determine how many days the model had to predict (25).
Furthermore, this model adds three "most critical output" variables to the dataset: "yhat," "yhat lower," and "yhat upper." For each day, yhat indicates the average, yhat lower represents the lowest value, and yhat upper represents the highest value.
This model recognizes "change points" automatically (which represent abrupt changes in the forecast trajectory). This allows the trend to adjust to the predicted values appropriately.

Finally, we used the causal impact model as a final model to separate the causation from the correlation between the campaign and sales. It was feasible to quantify the impact (increase or decrease) of the campaign on sales using this methodology.

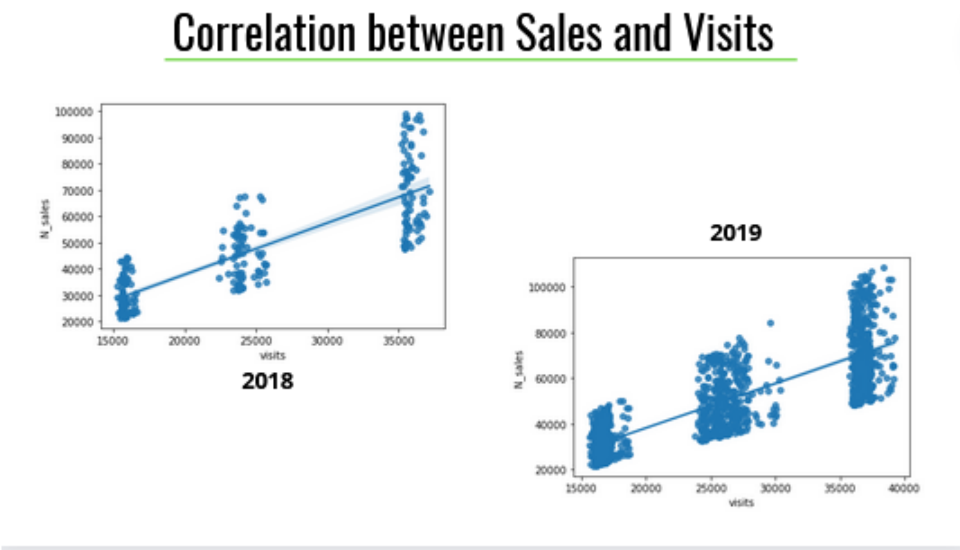
Results

For both 2018 and 2019, we looked at the relationship between visits and sales.
This **correlation** can be measured using the rho coefficient. It's seen through the array:
For 2018:

```
[[1.         0.81866719]
 [0.81866719 1.        ]]
```

For 2019:

```
[[1.         0.80501001]
 [0.80501001 1.        ]]
```



The correlation was utilized to investigate and show the linear relationship between the two variables of sales and visitors. Simple relationships can be described without needing to discuss cause and effect.
We can see from the graphical representations of the correlation that there is a higher density of points in correspondence with some values. In 2018, we can focus on values between 20000 and 40000, 40000 and 60000, 50000 and 1000000 for the variable "sales," and 15000, 25000, and 35000 for the variable "visits." In the year 2019, for the variable “sales” we can focus on values between 20000 and 42000, 23000 and 30000, 37000 and 40000; while, for the “visits” variables we focus on values between 15000 and 20000, 25000 and 30000, 35000 and 40000.

Granger Causality:
We decided to perform the Granger test statistic in order to know if visits cause more sales.
Null Hypothesis (H0) : N_sales do not granger cause visits. Alternative Hypothesis (HA) : N_sales granger cause visits.
Results: p-value is high, Null hypothesis is accepted hence N_sales are not granger causing visits. That implies visits came first.
Now repeat the Granger causality test in the opposite direction. Null Hypothesis (H0) : visits does not granger cause N_sales. Alternative Hypothesis (HA) : visits granger causes N_sales. The p-value is considerably low thus visits are granger cause N_sales. The above analysis concludes that the visits came first and not the N_sales. So more visits, more sales.

Multiple Linear Regression:
The multiple linear regression values for Milan 2019 are:

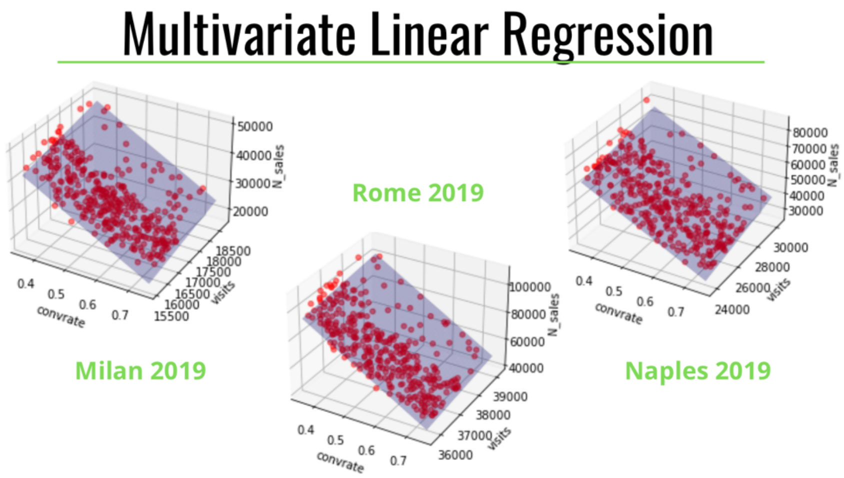
```
Parameters:  Intercept     74633.545609
convrate    -133067.085244
visits            1.866605
R2:  0.9611456298532879
```

For Rome 2019 the results are:

```
Parameters:  Intercept    33023.690284
convrate    -60307.315600
visits           1.905842
R2:  0.9597689235543408
```

For Naples 2019 we find:

```
Parameters:  Intercept    43751.300850
convrate    -82725.239341
visits           1.973172
R2:  0.9621705366292052
```



Decision tree

Decision tree for Milan 2019:

```
Accuracy level:  0.9975813249400057
```

	Real Values	Predicted Values
0	100229.946524	98885.026738
1	73761.224490	74451.345756
2	65913.996627	65098.333333
3	97806.970509	97885.333333
4	61947.107438	65098.333333
...	...	...
105	87871.046229	88434.146341
106	63018.675722	62979.557070
107	69771.062271	67660.617060
108	62415.692821	64041.166381
109	74141.716567	73956.175299

Decision tree for Rome 2019:

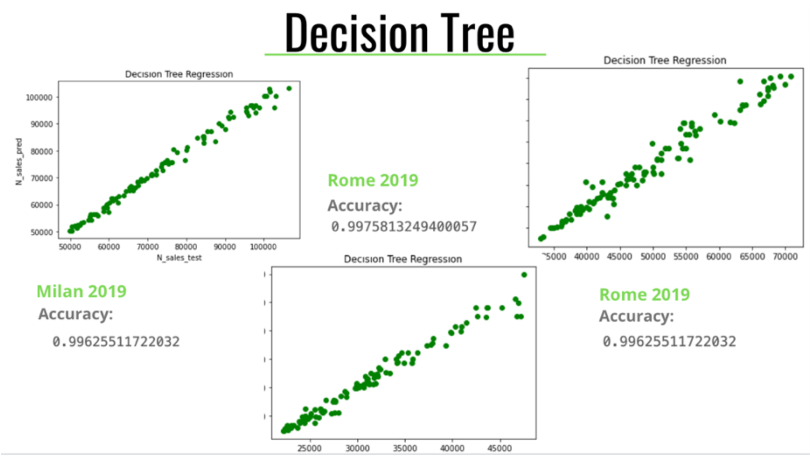
Accuracy level:0.99625511722032

	Real Values	Predicted Values
0	39556.074766	39627.039627
1	22229.166667	22480.821918
2	23808.035714	23760.772660
3	23314.002829	24129.814551
4	36864.864865	37122.448980
...	...	...
105	25568.778980	25629.860031
106	31940.663176	31518.771331
107	28857.885615	29022.569444
108	23029.126214	23061.983471
109	30206.349206	29792.321117

Decision tree for Naples 2019:

Accuracy level: 0.9932092906891082

	Real Values	Predicted Values
0	74647.696477	73426.229508
1	67382.151030	63305.045872
2	70038.043478	70030.054645
3	37348.765432	38412.044374
4	59628.318584	57019.736842
...	...	...
105	45363.957597	46653.778559
106	50736.220472	50891.304348
107	46086.058520	44727.879800
108	63918.854415	61307.328605
109	39091.463415	38282.738095



Time series models: ARIMA

In Arima, we used as variables firstly visits and then sales. Since from our measurements (carried out with the Dickey Fuller model and Rolling Statistics) the variable "visits" resulted to be non stationary with seasonality we transformed it using the logarithm into a stationary variable and eliminated seasonality. Rolling Statistics was recalculated in order to verify if the aim was achieved. The training set for this model takes data from November 2018 to October 2019. (the month before the campaign began). The data for the test set was computed between November 2019 and February 2020. This is done so that the model can forecast the values of the test set without take into account the campaign's impact.

The three order parameter values needed for Arima model were discovered through the autocorrelation plot. The model fit was then done. We restored the original scale to the results (since they were previously logarithmically transformed).

Finally, we used the error measure to visualise the model (RMSE).

ARIMA for visits Milan

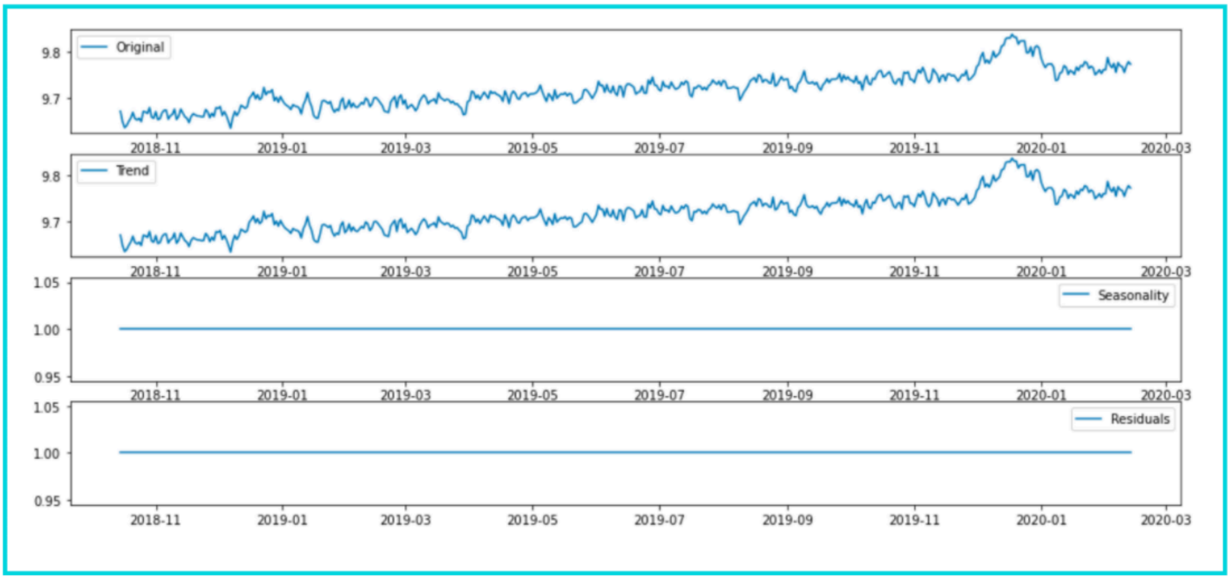
```
Milan, main features for visits:
     count       487.000000
     mean      36775.940452
     std         816.343609
     min       35213.000000
     25%       36189.000000
     50%       36696.000000
     75%       37284.500000
     max       39224.000000
```

Checking Stationarity for Milan

Method 1: Dickey Fuller

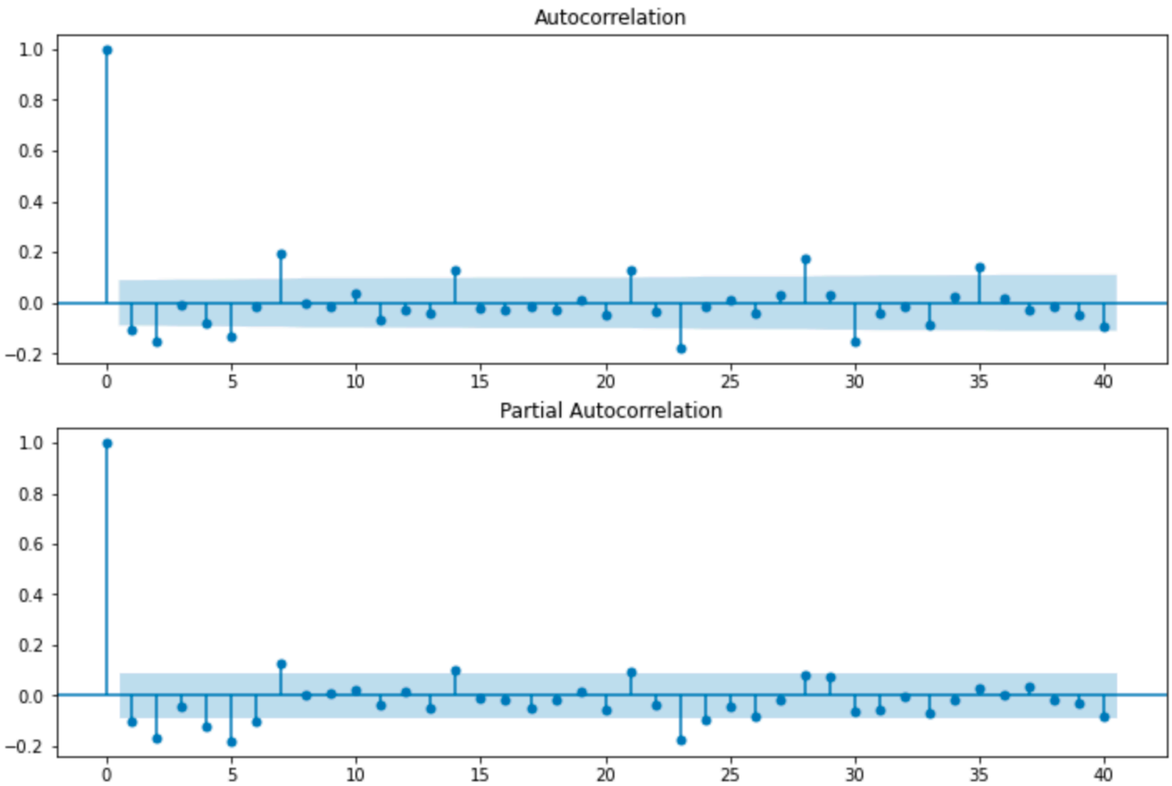
```
ADF Statistics: -1.5001017056485213
p- value: 0.5334610259801258
weak evidence against null hypothesis, time series has a unit root, indicating it
is non-stationary
```

`Method 2: Rolling Statistics`



Making series stationary with log transformation, Seasonal Decomposition, Differencing plot.

Autocorrelation plot: in order to select best parameters for the model.



ARIMA prediction:

```
day
2018-10-16    0.000112
2018-10-17   -0.000083
2018-10-18    0.000293
2018-10-19    0.001714
2018-10-20    0.003361
```

Arima log predictions:

```
                  day
2018-10-15    10.486262
2018-10-16    10.486374
2018-10-17    10.486179
2018-10-18    10.486554
2018-10-19    10.487976
```

```
RMSE: 477.1916
Finding the best parameters:
Best model:  ARIMA(3,0,0)(0,0,0)[0] intercept
Total fit time: 2.287 seconds
```

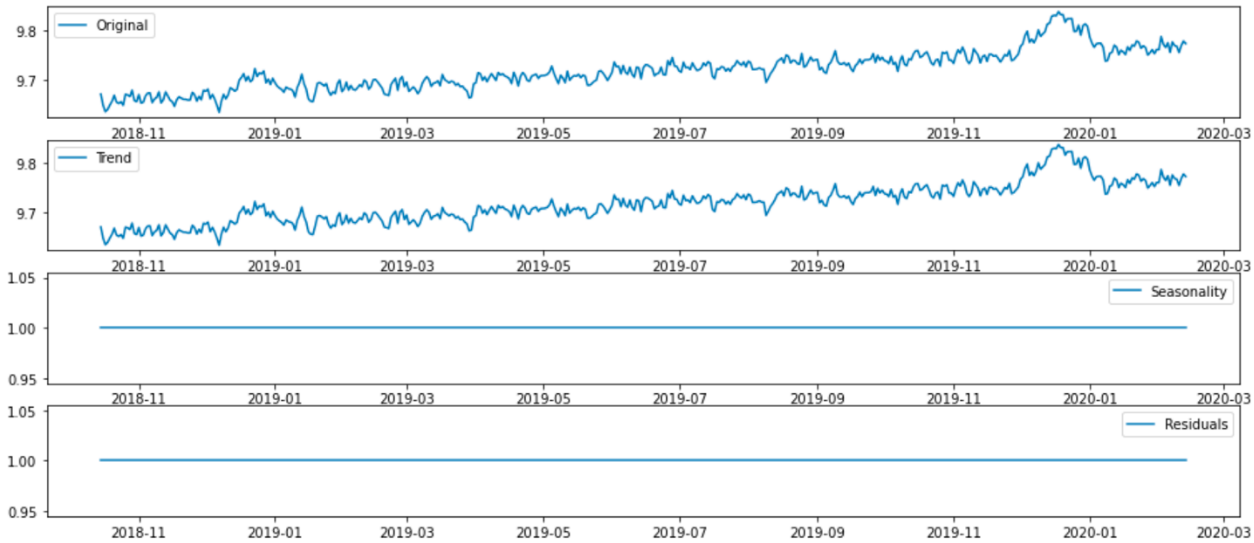
```
Rome, main features for visits:
     count        487.000000
     mean       16625.453799
     std          665.093469
     min        15251.000000
     25%        16166.000000
     50%        16574.000000
     75%        16996.000000
     max        18744.000000
```

ARIMA model for visits Rome

Checking Stationarity

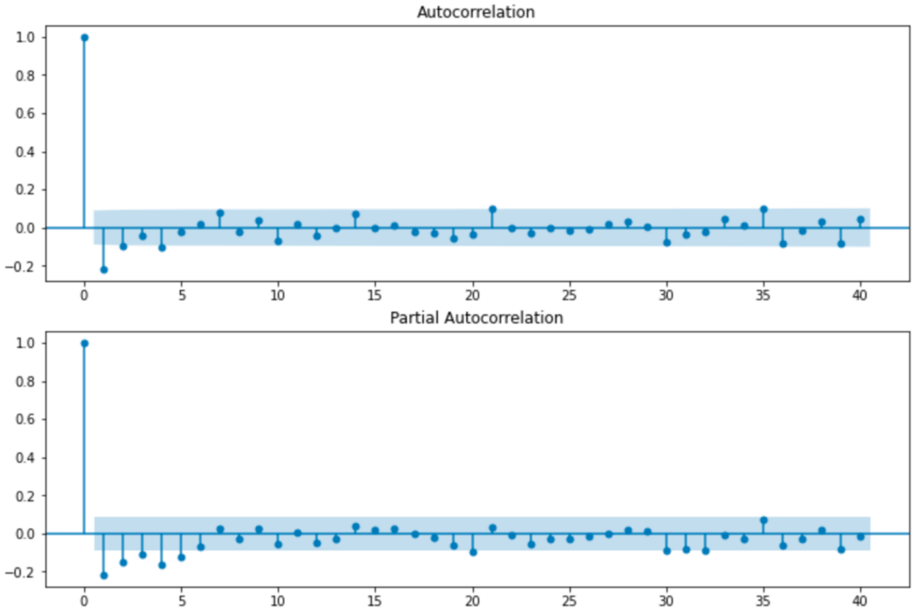
Method 1: Dickey Fuller

```
ADF Statistics: -1.5024102968637938
p- value: 0.5323186643253375
weak evidence against null hypothesis, time series has a unit root, indicating it
is non-stationary
```



Making series stationary with log transformation, Seasonal Decomposition, Differencing plot.

Autocorrelation plot: in order to select best parameters for the model.




```
Fitting the model
Predictions:
day
2018-10-16    0.000233
2018-10-17    0.005472
2018-10-18    0.012677
2018-10-19    0.016785
2018-10-20    0.015363

Log predictions:
day
2018-10-15    9.669662
2018-10-16    9.669895
2018-10-17    9.675134
2018-10-18    9.682339
2018-10-19    9.686447
```

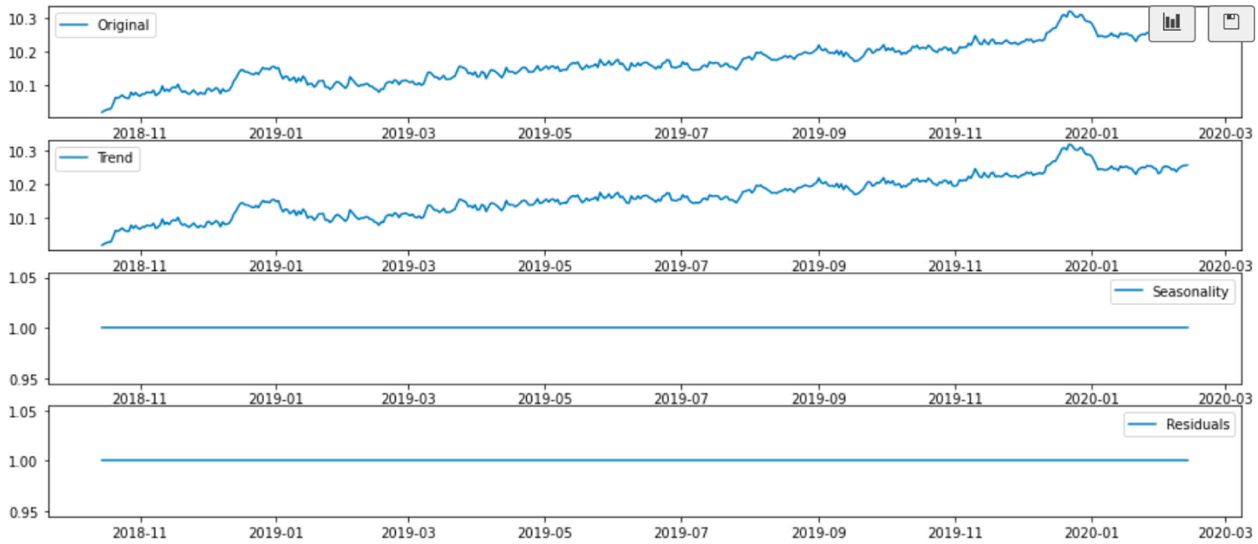
ARIMA model for Naples visits

```
General characteristics:
count      487.000000
mean     26013.223819
std       1577.805534
min      22428.000000
25%      24797.000000
50%      25776.000000
75%      27117.000000
max      30359.000000
```

Checking stationarity

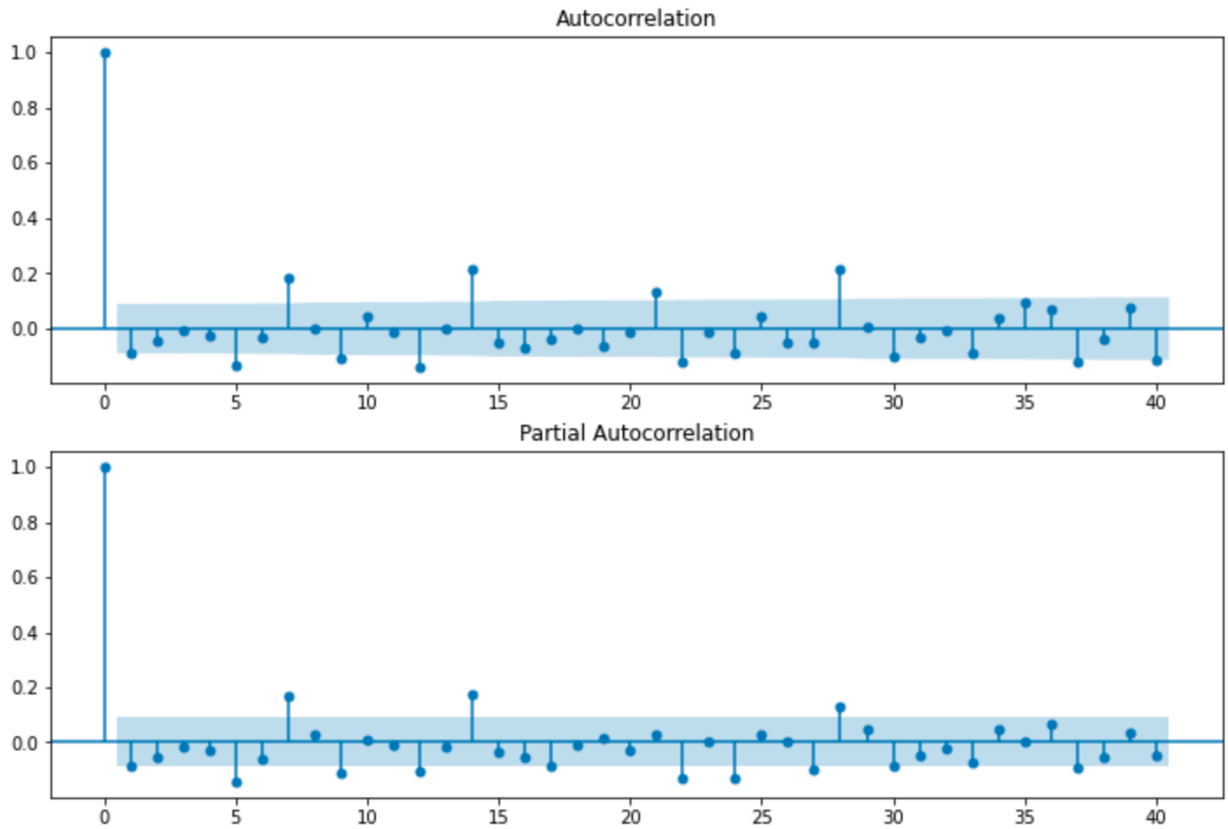
Method 1: Dickey Fuller

```
ADF Statistics: -1.135749480389806
p- value: 0.7006069827375822
weak evidence against null hypothesis, time series has a unit root, indicating it
is non-stationary
```



Making series stationary with log transformation, Seasonal Decomposition, Differencing plot.

Autocorrelation plot: in order to select best parameters for the model.




```
Fitting the model
                                       Predictions:
                                           day
                                   2018-10-16    0.000493
                                   2018-10-17    0.000688
                                   2018-10-18    0.000770
                                   2018-10-19    0.000907
                                   2018-10-20    0.001210
```

```
Log predictions:
      day
2018-10-15    10.018065
2018-10-16    10.018559
2018-10-17    10.018754
2018-10-18    10.018836
2018-10-19    10.018972
```

RMSE: 1020.1087

Sales variable:
Milan:

```
count       487.000000
mean      71116.896400
std       15860.474596
min       47602.150538
25%       57249.110344
50%       68273.764259
75%       83378.614468
max      108293.785311
```

ARIMA for sales Milan

Checking stationarity

Method 1 - Dickey fuller

ADF Statistics: -13.677832642252373
p- value: 1.419129002486693e-25
strong evidence against the null hypothesis, reject the null hypothesis. Data has no unit root and is stationary

The RMSE for test data is 16876.71163872828

Rome:

```
count      487.000000
mean     31520.777644
std       6949.564812
min      21293.405114
25%      25799.233836
50%      30206.349206
75%      36099.002660
max      49906.849315
```

ARIMA for sales Rome

Checking stationarity

Method 1 - Dickey fuller

ADF Statistics: -20.879766098253803
p- value: 0.0
strong evidence against the null hypothesis, reject the null hypothesis. Data has no unit root and is stationary

The RMSE for test data is 7437.78904973205

Naples

```
count      487.000000
mean     49591.105100
std      11440.684710
```

```
min       31912.751678
25%       39883.336644
50%       47475.319927
75%       56964.312865
max       84068.181818
```

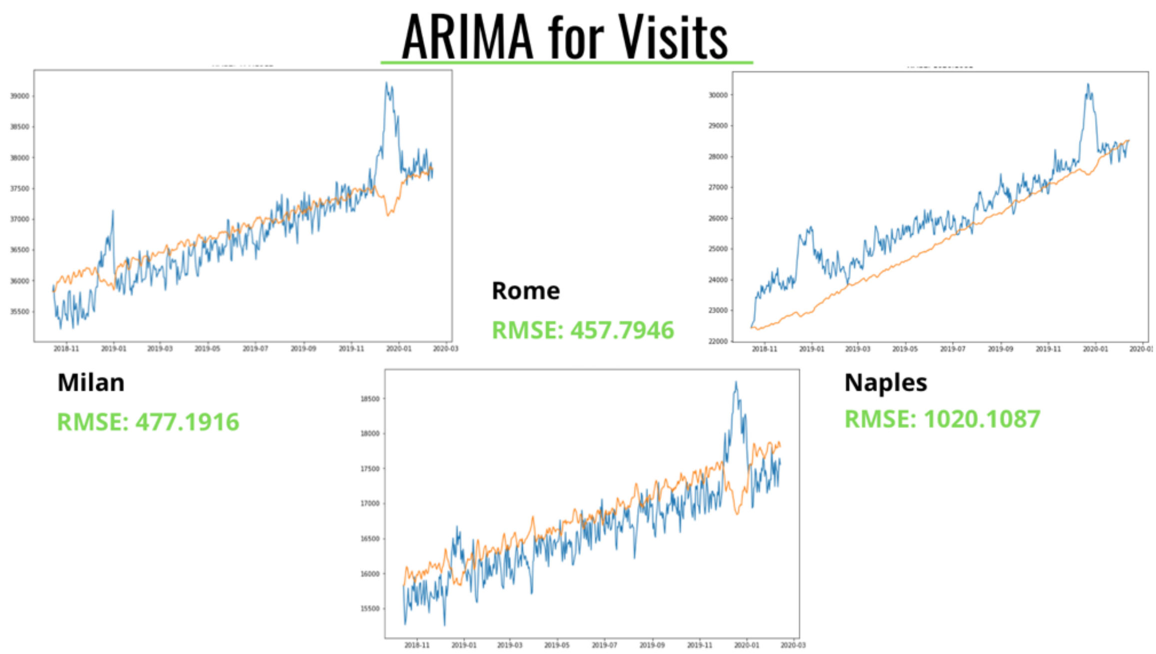
ARIMA for sales Naples

Checking stationarity

Method 1 - Dickey fuller

```
ADF Statistics: -19.53545093414264
p- value: 0.0
strong evidence against the null hypothesis, reject the null hypothesis. Data has no unit root and is stationary

The RMSE for test data is 12760.2318580992
```



The results for ARIMA visits are quite satisfying. With an average margin of error of 666 for the three cities, respect to the visits values range from 35 000 to 39 000. So, given the number of visits, the error isn't too bad. In terms of applying this model to sales, the results have been less than satisfying, since the error has an average value of 12 000 for a sales range of 40 000 to 80 000.

SARIMAX for visits

Sarimax is the second model that was used. This model can only be used for the variable visits because it is a non-stationary and seasonal model. The model's structure is identical to Arima's, with the exception of one new parameter: seasonal order.

SARIMAX Results

Dep. Variable:	y	**No. Observations:**	382
Model:	SARIMAX(1, 1, 2)	**Log Likelihood**	-2459.752
Date:	Thu, 21 Apr 2022	**AIC**	4929.505
Time:	18:37:01	**BIC**	4949.219
Sample:	0	**HQIC**	4937.326
	- 382		
Covariance Type:	opg		

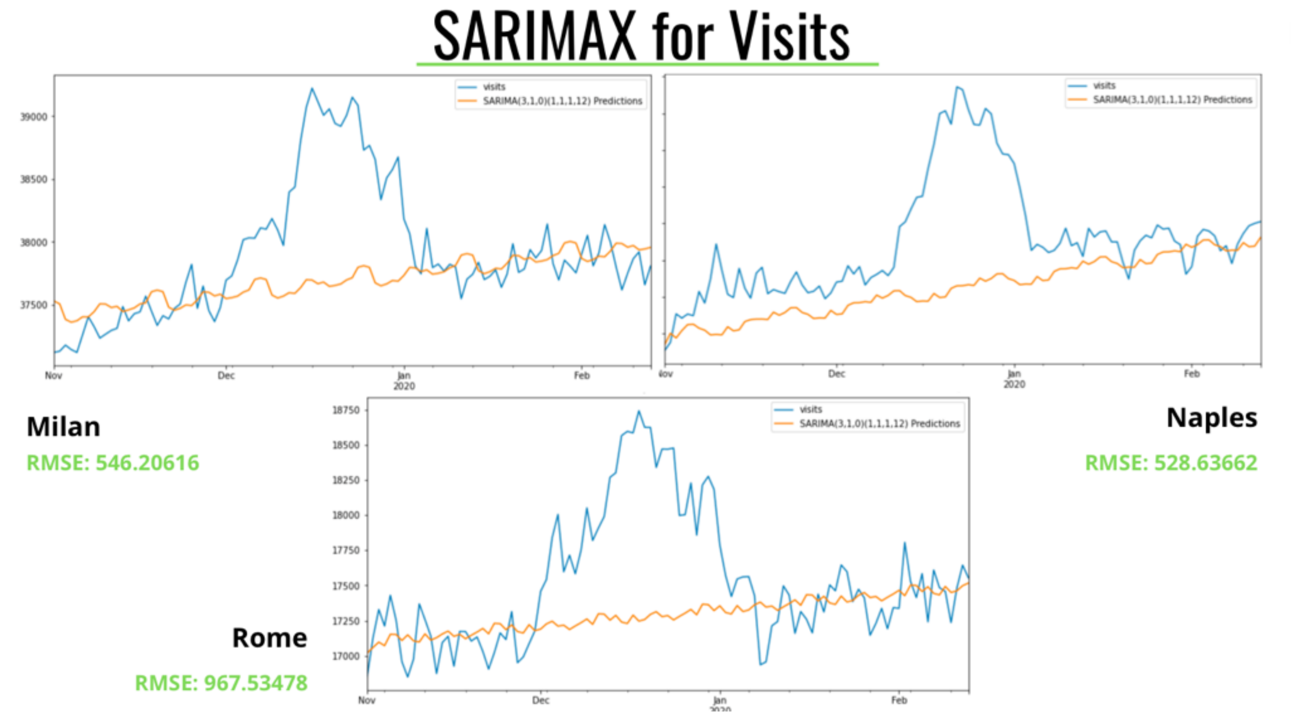
	coef	std err	z	P>|z|	[0.025	0.975]
intercept	2.7499	1.229	2.238	0.025	0.342	5.158
ar.L1	0.4820	0.098	4.917	0.000	0.290	0.674
ma.L1	-0.6773	0.102	-6.628	0.000	-0.878	-0.477
ma.L2	-0.1762	0.071	-2.473	0.013	-0.316	-0.037
sigma2	2.372e+04	1698.636	13.965	0.000	2.04e+04	2.71e+04

Ljung-Box (L1) (Q):	0.12	**Jarque-Bera (JB):**	0.89
Prob(Q):	0.73	**Prob(JB):**	0.64
Heteroskedasticity (H):	0.85	**Skew:**	0.04
Prob(H) (two-sided):	0.37	**Kurtosis:**	3.22

```
Milan
The RMSE for test data is 546.0512601295278

Rome
The RMSE for test data is 528.636622367324

Naples
The RMSE for test data is 967.5348192527642
```



This model's performance is satisfactory, with an average error of 680 for a range of values between 27 000 and 30 000.

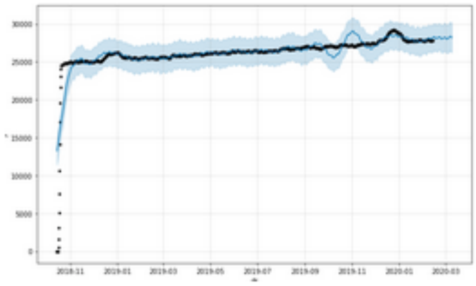
Facebook PROPHET for Visits variable

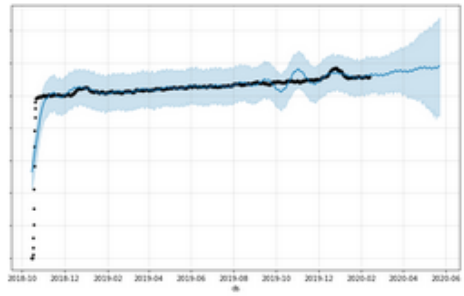
This is a model that permits to make high quality forecasts. First, we analyse the visits variable. It uses time series with some main model components that are combined: trend and seasonality. The input to Prophet is a dataframe with two columns (ds and y) and it returns a table with the following columns: **ds** (the timestamp entry of the forecast), **yhat_lower** (the bottom of the confidence interval for the forecast), **yhat_upper** (the bottom of the confidence interval for the forecast), and **yhat** (the forecasted value of the time series). These are the variables that gave us the predicted results with respect to the date specified.

Here we have the values:

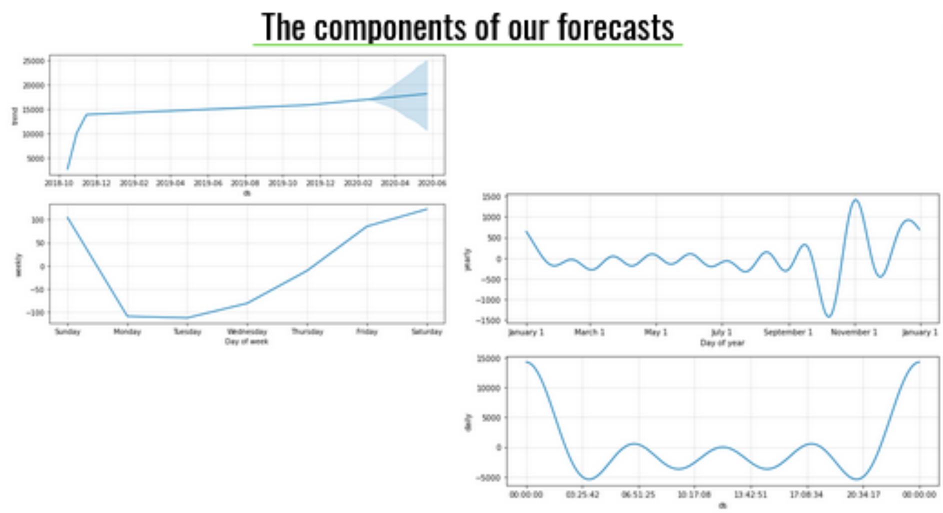
	horizon	mse	rmse	mae	mape	mdape	smape	coverage
0	4 days	146040.665127	382.152673	322.883547	0.011889	0.012293	0.011989	1.0
1	5 days	255228.521000	505.201466	478.094627	0.017582	0.018633	0.017757	1.0
2	6 days	397240.865336	630.270470	616.731454	0.022653	0.022554	0.022924	1.0
3	7 days	573370.429744	757.212275	744.025804	0.027306	0.026491	0.027698	1.0
4	8 days	889783.292284	943.283251	915.709396	0.033585	0.031797	0.034195	1.0

And here we have some graphical representations:





These two plots show the Prophet predictions; in the first one we did the prediction until March 2020, while in the second one we did the same prediction until June 2020.
As we can see in these plots, we are comparing predicted values and actual values. If the predicted values are lower than the actual ones, it means that the campaign has been successful. Looking at these results, we can say that the actual values are higher than the predicted ones: the champaign has been effective!
This model also permits to visualize the single components of our forecast, they represent the general trend until June 2020, and the weekly, yearly, and daily data:

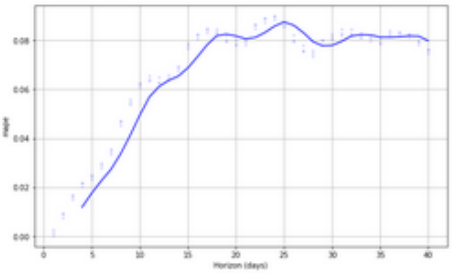


Then, we also computed some error measures to find the model accuracy:

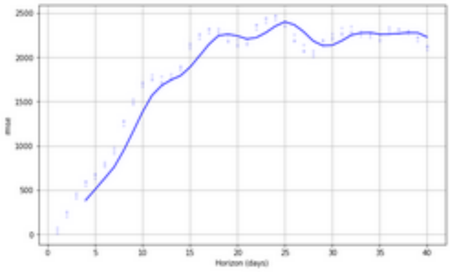
Error measures

	horizon	mse	rmse	mae	mape	mdape	smape	coverage
0	4 days	146040.665127	382.152673	322.883547	0.011889	0.012293	0.011989	1.0
1	5 days	255228.521000	505.201466	478.094627	0.017582	0.018633	0.017757	1.0
2	6 days	397240.865336	630.270470	616.731454	0.022653	0.022554	0.022924	1.0
3	7 days	573370.429744	757.212275	744.025804	0.027306	0.026491	0.027698	1.0
4	8 days	889783.292284	943.283251	915.709396	0.033585	0.031797	0.034195	1.0

MAPE



RMSE



We focused on the MAPE and the RMSE measures: they are very small, and it means that the accuracy of our model is good. We used the cross validation tool to generate the FBPROPHET error measurements, using the following settings: initial period: 365 days, period: 31 days, and horizon: 40 days.

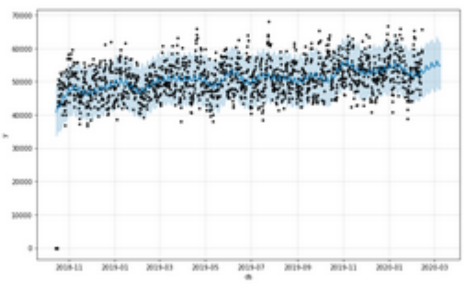
Facebook PROPHET for Sales variable

For “sales” variable we did the same forecasts:

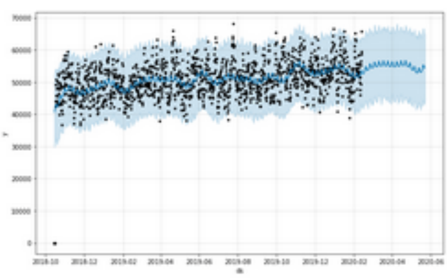
	horizon	mse	rmse	mae	mape	mdape	smape	coverage
0	4 days	2.486716e+07	4986.698482	4104.403242	0.076645	0.061006	0.072713	1.000000
1	5 days	1.470652e+07	3834.907951	3125.727129	0.055978	0.046282	0.054111	1.000000
2	6 days	1.935358e+07	4399.270335	3779.353250	0.068151	0.057738	0.065690	1.000000
3	7 days	2.844114e+07	5333.023296	4685.972167	0.086988	0.084602	0.082800	1.000000
4	8 days	6.626907e+07	8140.581548	6897.712712	0.139844	0.115759	0.126902	0.916667

These are the forecasts:

FBPROPHET for sales

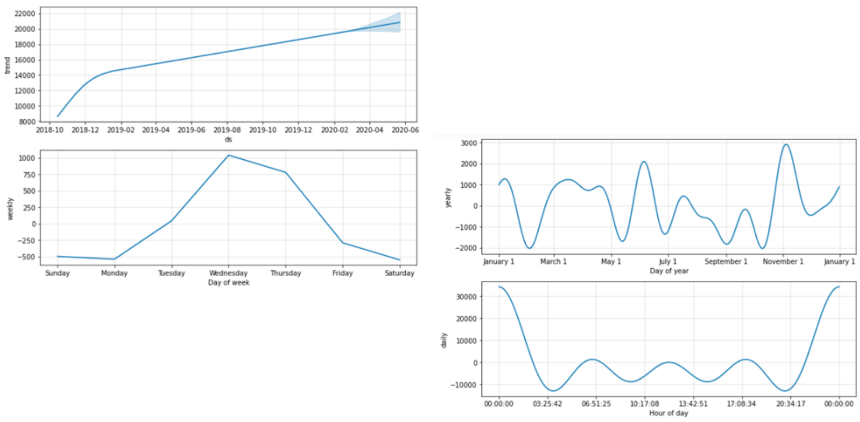


	ds	yhat	yhat_lower	yhat_upper
0	2018-10-15	40766.387876	33478.896761	48347.439285
1	2018-10-16	41627.411666	34244.913984	48872.669664
2	2018-10-17	42933.302498	35909.246550	49875.771850
3	2018-10-18	43010.530239	35450.177165	50476.922137
4	2018-10-19	42304.060715	35197.381986	48982.370324



The predicted components are:

The components

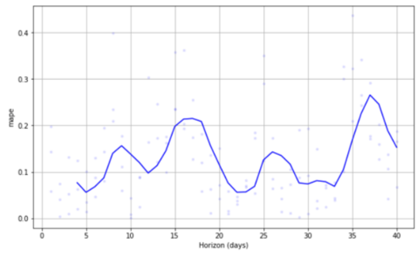


The error measures are:

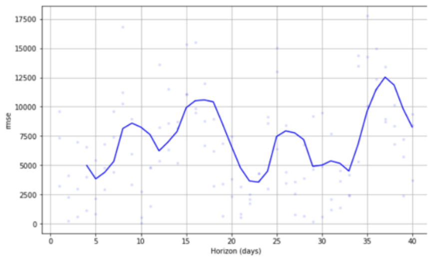
Error measures

	horizon	mse	rmse	mae	mape	mdape	smape	coverage
0	4 days	2.486716e+07	4986.698482	4104.403242	0.076645	0.061006	0.072713	1.000000
1	5 days	1.470652e+07	3834.907951	3125.727129	0.055978	0.046282	0.054111	1.000000
2	6 days	1.935358e+07	4399.270335	3779.353250	0.068151	0.057738	0.065690	1.000000
3	7 days	2.844114e+07	5333.023296	4685.972167	0.086988	0.084602	0.082800	1.000000
4	8 days	6.626907e+07	8140.581548	6897.712712	0.139844	0.115759	0.126902	0.916667

MAPE



RMSE



For this variable, the MAPE and RMSE are higher than the "Visits" variable, but the results are still good, and the model is accurate enough.

CAUSAL IMPACT sales

The examination of causation and correlation received special emphasis. Mostly because if data at hand were in fact generated by ad managers who increase ad expenditure in periods of the year when the purchase probability is higher, then the observed relationship between ad campaigns and sales could be highly predictive for the historical data but not useful in predicting the causal impact of explicitly assigning additional resources to an ad campaign.

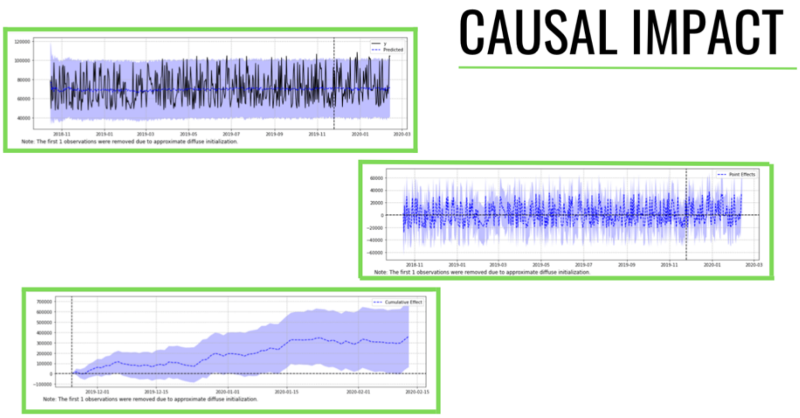
We performed a structured time-series analysis with causal inference by Google to determine if the ad campaign was the cause of an increase in sales during the same period. Causal inference is about determining the effect of an event or intervention on a desired outcome metric.

The first plot shows the actual sales (y) versus the predicted one in the whole period considered. The second plot shows the difference between the actual series and the predicted series, referred to as the point effects.

The third plot shows the cumulative effect, which is basically the summation of the point effects accumulated over time.

The model summary output presents the table with the values of the prediction and the relative effect of the campaign in absolute value (4534.44) and the relative effect (6.44%)

The causal effect, unlike correlation, proves that something has happened or is happening because of something else.




```
Summary:
Posterior Inference {Causal Impact}
                          Average            Cumulative
Actual                    74988.23           5999058.37
Prediction (s.d.)         70453.79 (2053.96) 5636302.84 (164316.92)
95% CI                    [66135.9, 74187.28][5290871.82, 5934982.29]

Absolute effect (s.d.)    4534.44 (2053.96)  362755.53 (164316.92)
95% CI                    [800.95, 8852.33]  [64076.07, 708186.55]

Relative effect (s.d.)    6.44% (2.92%)      6.44% (2.92%)
95% CI                    [1.14%, 12.56%]    [1.14%, 12.56%]

Posterior tail-area probability p: 0.01
Posterior prob. of a causal effect: 99.2%
```

REPORT:

```
Analysis report {CausalImpact}


During the post-intervention period, the response variable had
an average value of approx. 74988.23. By contrast, in the absence of an
intervention, we would have expected an average response of 70453.79.
The 95% interval of this counterfactual prediction is [66135.9, 74187.28].
Subtracting this prediction from the observed response yields
an estimate of the causal effect the intervention had on the
response variable. This effect is 4534.44 with a 95% interval of
[800.95, 8852.33]. For a discussion of the significance of this effect,
see below.
```

```
Summing up the individual data points during the post-intervention
period (which can only sometimes be meaningfully interpreted), the
response variable had an overall value of 5999058.37.
By contrast, had the intervention not taken place, we would have expected
a sum of 5636302.84. The 95% interval of this prediction is [5290871.82,
5934982.29].


The above results are given in terms of absolute numbers. In relative
terms, the response variable showed an increase of +6.44%. The 95%
interval of this percentage is [1.14%, 12.56%].


This means that the positive effect observed during the intervention
period is statistically significant and unlikely to be due to random
fluctuations. It should be noted, however, that the question of whether
this increase also bears substantive significance can only be answered
by comparing the absolute effect (4534.44) to the original goal
of the underlying intervention.


The probability of obtaining this effect by chance is very small
(Bayesian one-sided tail-area probability p = 0.01).
This means the causal effect can be considered statistically
significant.
```

Discussion and/or Conclusions

It can be concluded that the advertising campaign had a 6.44% favourable impact on sales. Despite the fact that the period in which it was executed was marked by a rise in sales, the analyses conducted show that the increase would not have been as great if the advertising campaign had not been implemented. Furthermore, all three models utilised yielded a good outcome, proving this claim. The use of three models was done to ensure that the results were significant.

References

https://towardsdatascience.com/time-series-analysis-with-facebook-prophet-how-it-works-and-how-to-use-it-f15ecf2c0e3a

https://www.analyticsvidhya.com/blog/2021/08/granger-causality-in-time-series-explained-using-chicken-and-egg-problem/

https://towardsdatascience.com/practical-python-causality-econometrics-for-data-science-ffc074e11a1d

https://github.com/nickdcox/ml-causal-inference-swe/blob/main/Causal%20Inference%20for%20Bug%20Reporting.ipynb

https://blog.exploratory.io/an-introduction-to-causal-impact-analysis-a57bce54078e

https://nextjournal.com/fb-prophet/facebook-prophet-diagnostics

https://stephenallwright.com/good-mape-score/