

Master Project Report

Forecast Sale Model



Introduction

I have been working in the Digital Industry more than 13 years and in the Sales department (eCommerce) more than 11 years and during those years there was something rambling on my mind... How can we forecast our sales? This question accompanied myself across all my career and I have tried using 'standard' tools like excel and Regressions for giving an answer.

One of the main questions that any people who work in a management department is... how am I performing versus market? Understanding the addressable market, our competitors and our performance could give us a measure of how good we are doing and even more important. Set targets for upcoming years. Normally, if you work in the Airline Industry you can use Accelya in order to understand how good you are doing versus market but you cannot know how good your eCommerce is doing versus other airline websites. If you work in the Retail Industry you can use GFK, IDC, Nielsen and so on. Those market research companies could give you a very good picture about your performance across vary variables like price bands, products, volume, value and so on... but again, corporate eCommerce is not there. It is not tracked on those tools. So, how can we set targets if we do not know the addressable market or how my competitors are doing?

On the other hand, one of the most important key success factor in any business is the price of your product or service. Normally, you segment your clients or market by business. You have different route to markets and therefore different teams who tackle those clients. How does eCommerce fit in this 'old-paradigm'? And how eCommerce and the organization itself would deal with the famous 'Channel Conflict'.

These are the main two germens which explain my project. Assuming organization does not care too much about market and the idea of gaining market share and therefore more revenue, what I pretend is to create a model which forecast my upcoming sales in a timespan of two weeks in order to answer this important question. Considering my forecast based on my current promotional activities, stock level and demand generation investment, would we hit our weekly targets? If so, let's continue business as usual but if this model tells us that we need to do more, this will help us to work with the different teams to get better promotions and logically manage the channel conflict much more wisely. We are going to run(this model will not tell how much) more promotions or increase the traffic to the site in order to hit our weekly targets. Therefore, we will have a tool which help us to expand the business minimizing the internal frictions between the traditional business and the new business, the eCommerce.

Another important decision which has me pushed towards this project was that I have not seen anything like this any other company. Across my extend experience in this field and having worked in one of the most important leaders in their sectors, I have not seen anything like this before. Probably due to the project's complexity or maybe because the people who have faced this challenge did not have the right skills to develop it.

Therefore my project is around forecast the eCommerce sales in the Retail Industry. Specially in the IT industry.

Environment and data sets

I have used the Virtual Kschool Machine with the Anaconda distribution and Python

Python Packages required:

Typology	Library
Array and Frames libraries	numpy pandas
Plotting libraries & Visualization	matplotlib seaborn altair streamlit Image
Analyze distributions	fitter
Machine Learning libraries	sklearn xgboost lightgbm
Alternative Machine and Deep learning libraries	prophet tensorflow & keras
Statistical Models	statsmodels patsy
Regular Expressions Libraries	re
Avoid warnings	warnings
Progress Bar	tqdm
Serilizing	pickle
Combination	tertools
Dates	datetime relativedelta

Pip instructions:

- pip install fitter
- pip install xgboost
- pip install lightgbm
- pip install pystan==2.19.1.1
- pip install fbprophet
- pip install tensorflow
- pip install keras
- pip install streamlit

There are two main data sets which have not been included in GitHub and they have been sent separately by email to Dani and Igor:

- data_v1.xlsx – This is the main set that I would use for training and testing
- data_validate_v1.xlsx – This will be the set for validating the trained models

Those two files should be in the same folder as the rest of the files of the project.

Data Sources

Any Data Analysis require to have clean data and that is other of my hobbies. It is not my responsibility but in somehow, I have been one of the people in charge of defining how to classify our products because I have always had a clear objective. All previous data clean jobs should have been done before I started my project in order to try forecasting our sales. So, that phase is not done at 100% but it looks like in good shape.

Across my experience in my current position, besides trying to have as much cleaner data as possible, I think I have spent quite remarkable time identifying the different data sources.

I am going to use one main data source to face this part:

- Web Tracking Source: We are using best-in-class tool for this purpose and I master it quite well. Here I can find a lot of data from different granularities

Data Description and Dictionary

These are type of data that I am going to use for this project:

Date fields

- Date: When the transaction has been made
- Program: What traffic source was attributed for Sale or Traffic
- Visits: Amount of visitors
- Revenue: Amount of money triggered by the sale

When I tackled this project, I thought to add more information as stock level, type of product, shipping dates and so on but teachers suggest me to treat this problem as it were a Time Series. Therefore, I am not going to use so many endogenous variables, just Date and Revenue. I will avoid Program and I will keep Visits in case I would like to lag those ones as well.

Project execution

This project has been grouped in four type of files:

- Notebooks: Just focus on the Data Science analysis from the beginning until the end
- .py file: All functions have been located here
- Excel files: As we saw above, these files are the data for this project
- Icon: This is an icon used for the frontend

All files should be downloaded in the same folder because all references are based on current folder.

Forecast_Sale_Model.ipynb

This is the main notebook. For replicating purpose, just you should follow the same cell order and execute them. Just one comment, there are some Grid Search for Time Series quite intense in terms of computation effort, so be warned those cells could last some hours.

Forecast_Sale_Model_Visualization.ipynb

This notebook is aimed to create a **visualization.py** file for visualization purpose. I am using streamlit

functions.py

This file has all functions and methods for our project. I have tried to make all functions as much as modular as possible

visualization.py

This file is the output from **Forecast_Sale_Model_Visualization.ipynb** and it is the input for the streamlit server

Tools and methodology

All phases of this project has used Python, data gathering, data manipulation, exploratory data analysis, classical methods, machine learning models and optional models. Finally, our fronted has been programmed with Python and exploiting by streamlit.

I think, the biggest challenge that I faced during this project, it was classifying properly what category problem is. For example, at the beginning I was spending some time collecting stock information to add those features in my machine learning model. However, I was guided wisely that my project belongs to Time Series category.

Once I understood what type of problem I wanted to resolve, I spent some time understanding Classical Models, Machine Learning limitations and finally I added some optional two models like Prophet and RNN.

Before to move towards the used models, the data which is one critical part of this project was not the main issue because in general, I just need to lag them. That was another important learning

From the Classical Models, my main challenge was understanding how an ARIMA, SARIMA and SARIMAX models workout and the understanding behind them. This is my way of thinking, I need to understand how things work in order to use them properly. I spent some time programming ARMA algorithm from the scratch and that invested time was not wasted because that helped me to move towards the Statmodels library and use it in a better way.

In Machine Learning part, my main concerns were related to how we should treat our features and one important challenge. How I tackle a Cross Validation in a Time Series. According by Rob Hyndman, we can pick data and split between Train and Test just we need to ensure Test data is not included in the Train group. Reference: <https://robjhyndman.com/hyndsight/tscv/> and special thanks to Sebas for his support in this point.

In a Toni Almagro class, he explained us that we should use Date features as categorical because sometimes there are not linear dependency. I have decided to use my Date features in the same way thanks to Toni but also looking at the correlation between the target.

Lastly, I decided to try in a briefly way two methods: Prophet and RNN. All effort put in place helped us to quite straightforward testing those models where however, I have just scratched the surface and probably there are much more possibilities.

Data manipulation

The data is quite clean because it comes from a Web Tracking tool. Something to bear in mind is about the data quality. In general, all transactions tracked in this tool are confirmed but there are some which have been resolved from our Telesales team and sometimes the customers call us for the same order because this person wants to amend it but our systems do not allow us to do it and we need to replicate it in the system and there is no way to remove any activity made on the Web and therefore those amends could be double or triple counted. Besides this potential contamination in the data, sometimes we have suffered some outages in the platform but not greater than 6 issues for not greater than 24 hours in a period of 6 years. So, quite petty.

Main task done in this part was:

- Check those outages with 0 Revenue value
- Index the Dataframe with Date values
- Group by Visit and Revenue in order to avoid traffic sources name

	Year	Month	Day	Visits	Revenue
Date					
2021-05-05	2021	5	5	14373	37283.53
2021-05-06	2021	5	6	14025	39991.11
2021-05-07	2021	5	7	12399	24901.84
2021-05-08	2021	5	8	11018	15744.14
2021-05-09	2021	5	9	13924	19703.32

Exploratory Data Analysis

Despite of more than 6 years of data, we do not have so many rows, 'just' 2321

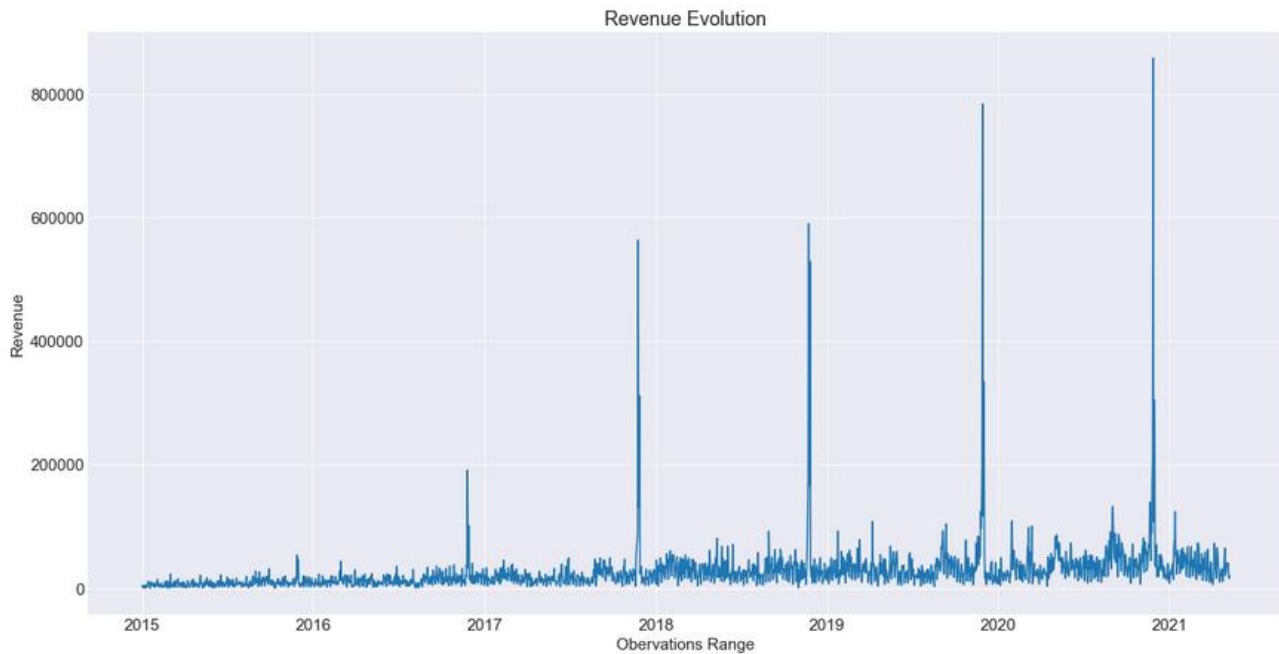
```
In [18]: #Lets check it out how this looks like and we can see that we have 2293 rows of data. Original one was 104793 rows
data_small.shape
```

```
Out[18]: (2321, 5)
```

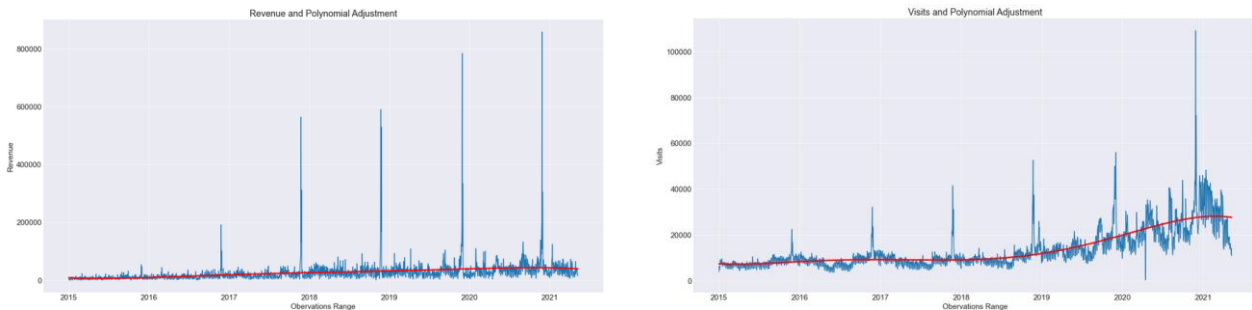
Basic statistical describe, I can see how fluctuate for example Visits and Revenue. Both shows outliers mainly happened during Black Friday week. I found 17 outliers in Revenue

	Year	Month	Day	Visits	Revenue
count	2321.000000	2321.000000	2321.000000	2321.000000	2321.000000
mean	2017.694959	6.308057	15.680310	13433.658337	25825.089417
std	1.843913	3.479171	8.807686	8616.702674	38526.622896
min	2015.000000	1.000000	1.000000	245.000000	46.420000
25%	2016.000000	3.000000	8.000000	7858.000000	9739.620000
50%	2018.000000	6.000000	16.000000	10223.000000	18558.480000
75%	2019.000000	9.000000	23.000000	16581.000000	32819.390000
max	2021.000000	12.000000	31.000000	109148.000000	858333.560000

This is how the data looks like. As we can see there are 5 peaks from Black Friday and Cyber Monday and their respective Saturdays and Sundays.



An interesting observation is how Revenue and Visits are adjusted with a simple polynomial and we can see that despite of those peaks, Revenue is quite 'stable' vs. Visits



Actually, our main target, Revenue is fulling one important behave needed for Classical Methods, called Stationary.

```
test_adf(data_small, 'Revenue')
```

**** Augmented Dickey-Fuller Test ****

T-test: -7.039862496454264 < Confidence Interval[1%]: -3.4332013179632686 - Result: Stationary

T-test: -7.039862496454264 < Confidence Interval[5%]: -2.862799647940788 - Result: Stationary

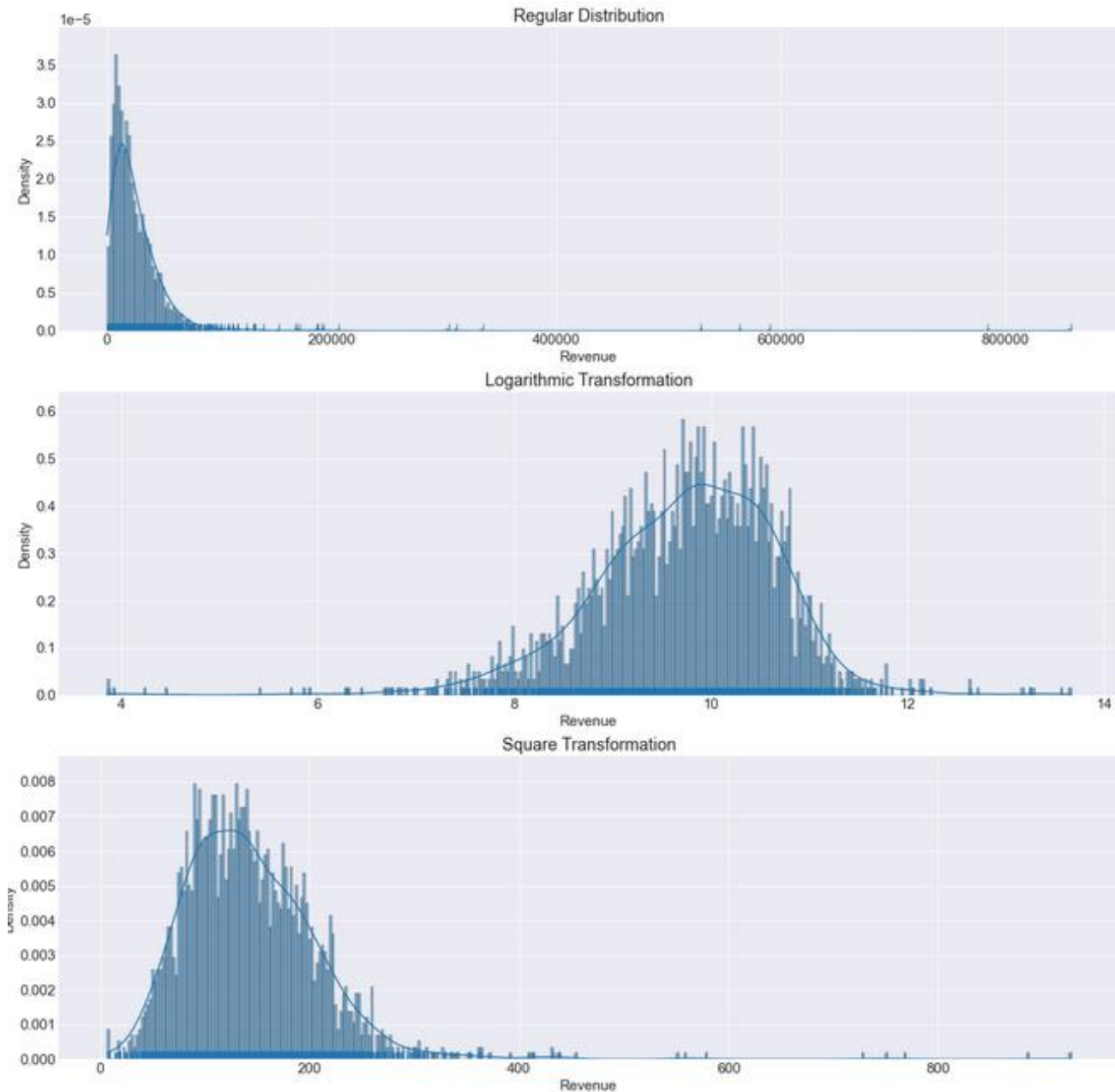
T-test: -7.039862496454264 < Confidence Interval[10%]: -2.5674405676968886 - Result: Stationary

P-Value: 5.884812935635475e-10 < 0.05 - Result: Stationary

As far as we can see here, it seems this Time Series is Stationary. If we check same Time Series by Week, everything changes because it seems Non Stationary.

Distribution Analysis

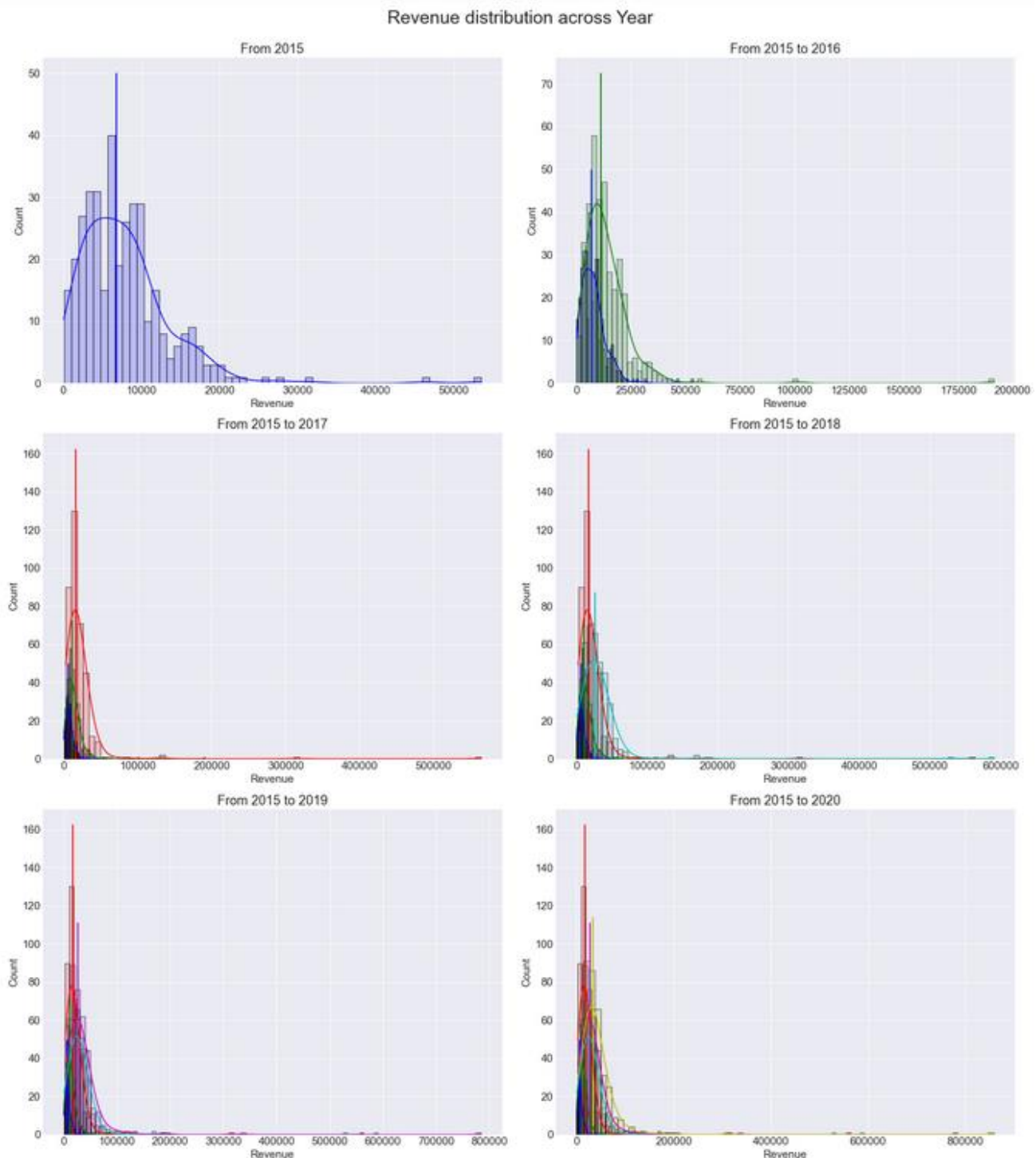
Another interesting analysis is about how the Revenue distribution looks like. I will show how the data is just 'raw' and manipulated also by Log and qrt.



As we can see Revenue variable shows kind of Gamma Distribution and thanks to applying Log and Sqrt transformation, we can see the distribution turns into Norm Distribution. Actually, if we apply the fitter library, we can see what Distribution fit better and in general Revenue follows a Log-Normal distribution.

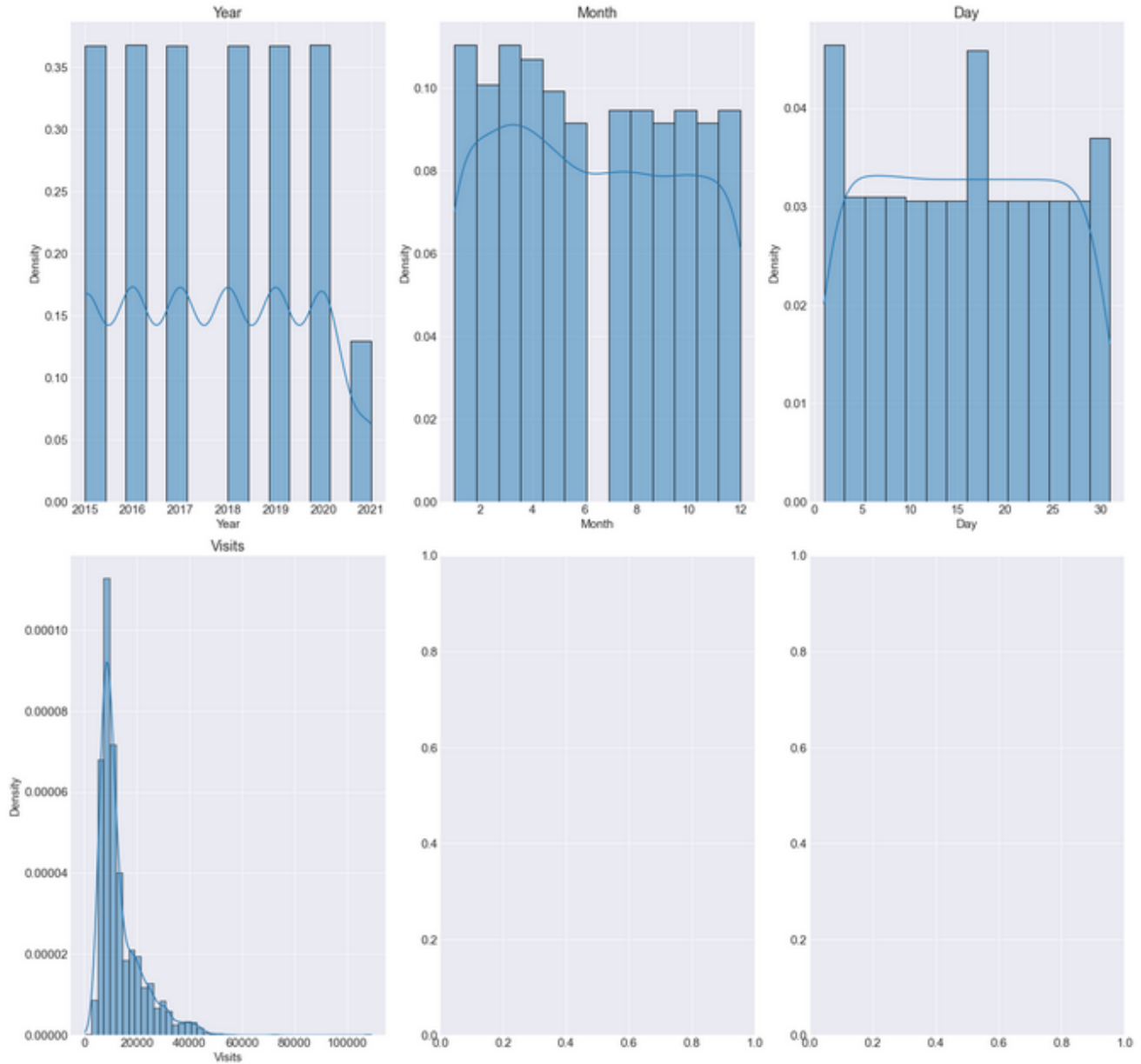
In terms of distribution, I wanted to analyze how this distribution shape evolve across the time and it is quite interesting. We can see that the tails are going further(outliners, this means Black Friday are much and much extreme than in the past) and on the other hand, 'regular sales' are much and much concentrated on the left

hand-side. I have tried to separate the histogram by median and I could see how data follows that Log-Norm distribution as we have seen before.



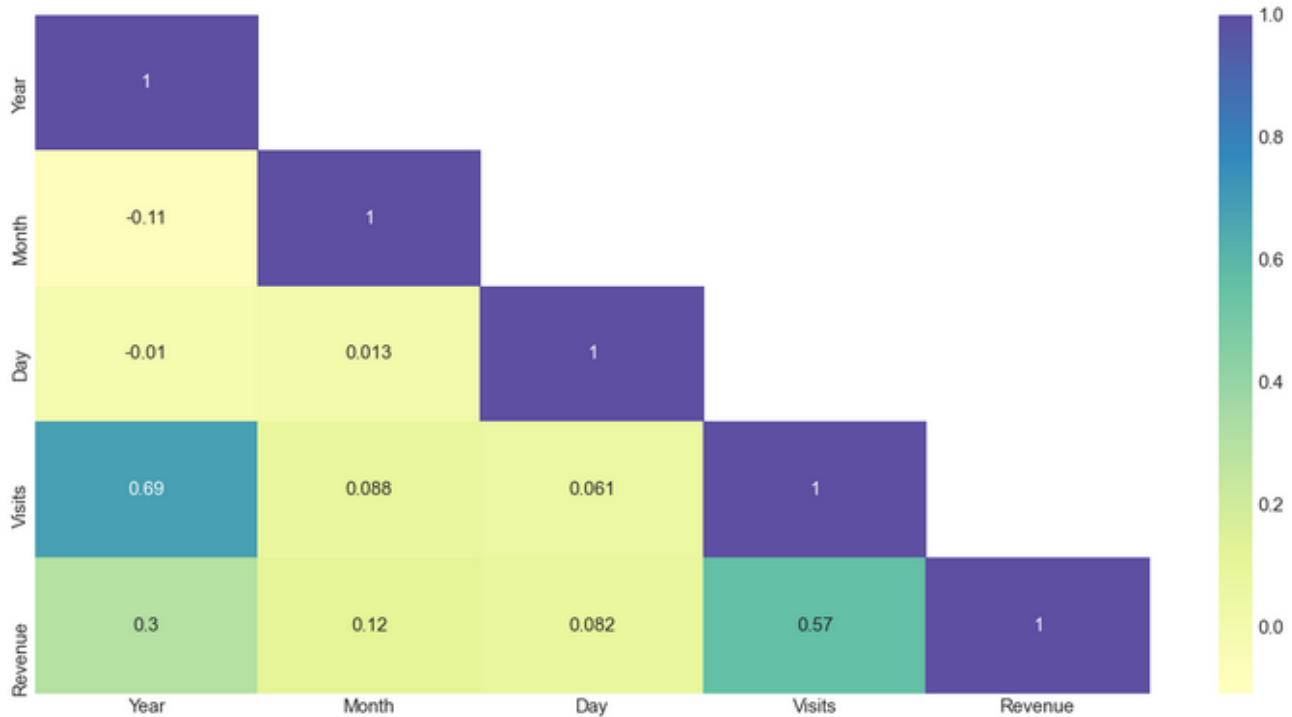
Checking how the features(in this analysis I have included Visits, but this is not used in our model) are distributed, we could see that Visits follow in somehow the Revenue distribution. Actually, if we pay attention about eComm Business, Visits probably is the most important feature for this business. Year, Month and Day follows the distribution that we could expect and therefore, we have decided to use them as categorical features.

Features Distributions



Correlations Analysis

As I mentioned before, data talks by itself. Visits is one feature highly correlated to Revenue, the second one is Year and that makes sense as we could see before. As long as the time goes by, Revenue grow more.



Featuring Engineering

Because I have used different approaches or models, this part is slightly different if we use Classical Methods than Machine Learning for instance.

Classical Methods:

- Exogenous features as categorical
 - Easter
 - Covid
 - Black Friday
- Target: Revenue

Machine Learning Methods:

- Features
 - Easter, Covid and Black Friday as seasonal and I treated them as if they were categorical
 - Lagged Revenue. We are going to use 8 lag levels [1, 2, 3, 4, 5, 6, 7, 364]. There is a week trend and there is a strong correlation with the same day versus last year



- Target: Revenue

Alternative methods:

- Prophet
 - I have used as SARIMAX, Easter, Covid and Black Friday as features and target Revenue. However, DataFrame should follow a specific format and quite different than the rest
- RNN
 - I have used as Machine Learning, Easter, Covid and Black Friday as seasonal features plus lagged Revenue at same scale as we used on those models and I used Revenue as target again

Models

Considering that we are trying to resolve a Time Series problem, I will tackle this project in three different groups:

- Classical Methods
 - SARIMAX
 - Triple Exponential Smoothing
- Machine Learnings Methods:
 - Linear models
 - Non-linear models
 - Ensemble models
- Alternative methods:
 - Prophet
 - RNN

Main metric used has been RMSE and the main objective is forecasting the Revenue in T+1 and so on. I have tried to construct prediction intervals in all methods. According by Rob Hyndman <https://otexts.com/fpp2/prediction-intervals.html> we can use the residuals if they follow a specific conditions to create those intervals. If residuals follow a Norm Distribution, we can use Rob's assumptions.

Before I will analyze the different models, I am going to use as a baseline the N  ive Model. Basically, lagging the target Revenue T-1. The RMSE is 40663

Classical Methods

SARIMAX

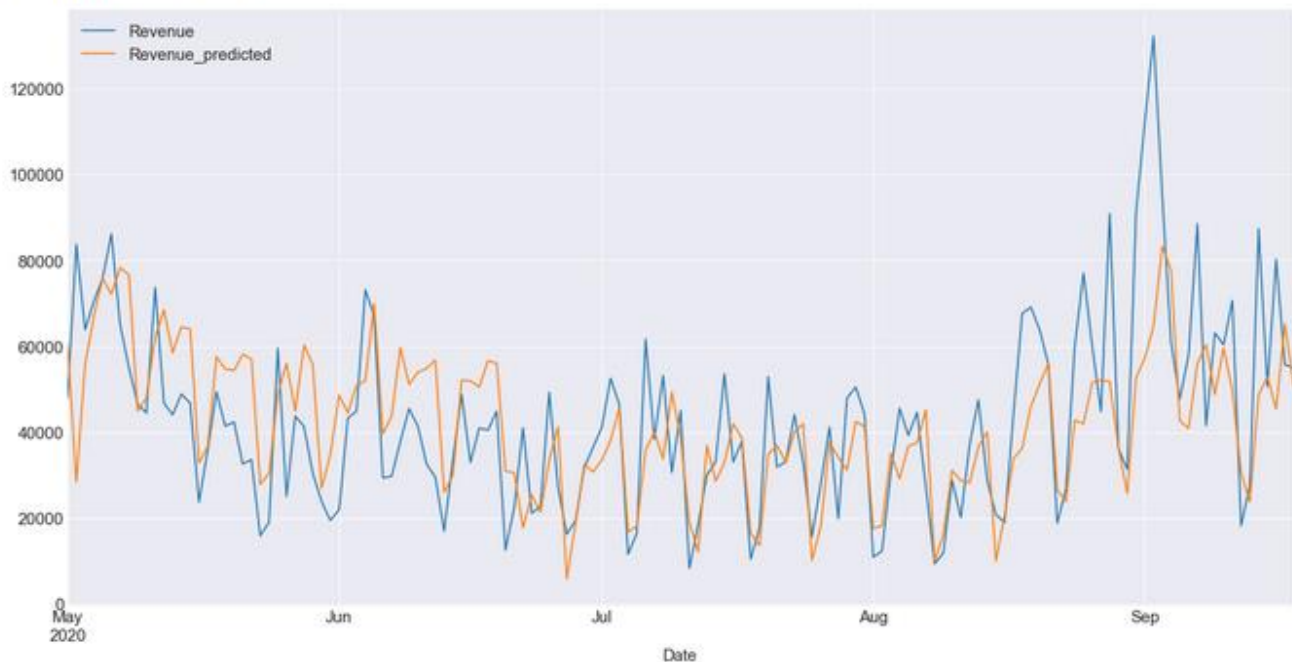
My main challenge here was creating a Grid Search algorithm from the scratch to get best hyperparameters. This is extremely computing intense and took several hours to perform. What I have done was divide the Grid Search in a subset, pickle the results and continue with another set of parameters and so on. Finally, I got best parameters.

```
#Fit considering best hyperparameters
model = SARIMAX(endog= y_train, exog= X_train[['Black_Friday', 'Easter', 'Covid']],
                order= (3, 0, 0), seasonal_order= (1, 0, 1, 7), trend= 't')
result = model.fit()
```

Results are quite impressive, showing strong RMSE way above Näive.

```
#Let's see a subset of the Time Series
data_train_predict.loc['2020-05-01':'2020-09-18'].plot()
```

<AxesSubplot:xlabel='Date'>



```
metric_rmse(data_train_predict, 'Revenue')
```

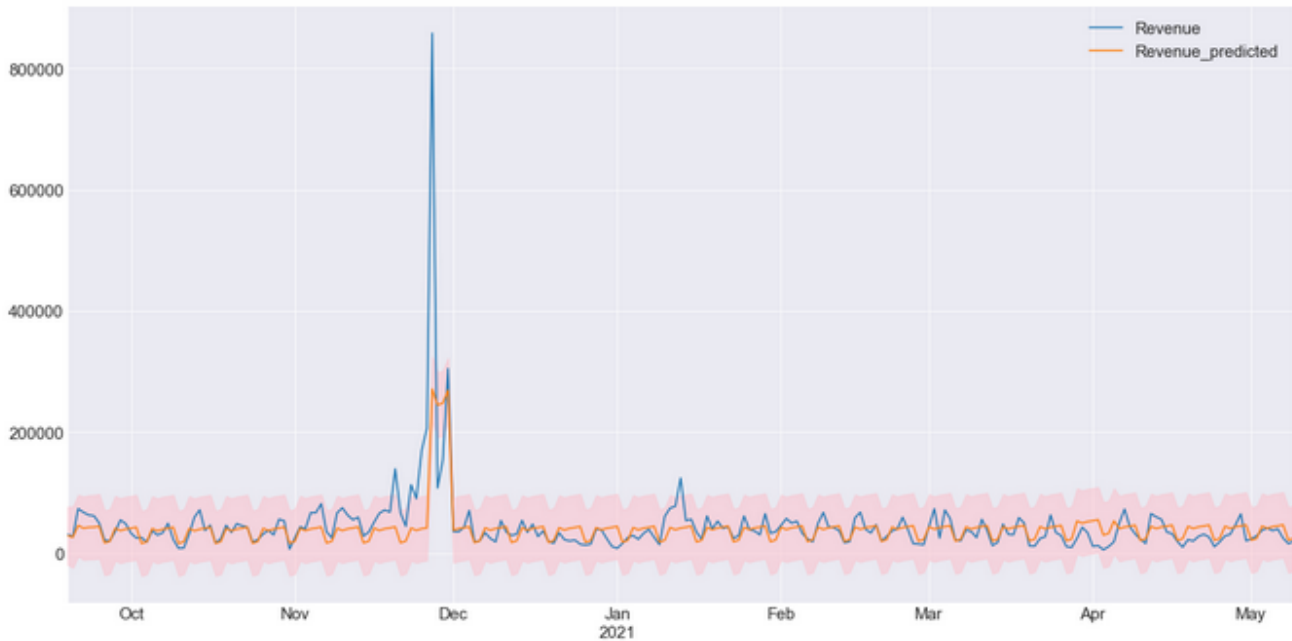
22941.71458906555

However, RMSE plumed strongly when we validate the model with the test data.

```
#Let's see how this model predict
data_test_predict = pd.concat([y_test, mean_forecast], axis=1)
data_test_predict.plot()
```

```
confidence_intervals = forecast.conf_int()
plt.fill_between(confidence_intervals.index, confidence_intervals[confidence_intervals.columns[0]],
                 confidence_intervals[confidence_intervals.columns[1]], color='pink', alpha=0.5)
```

<matplotlib.collections.PolyCollection at 0x7fe506a4b340>



```
metric_rmse(data_test_predict, 'Revenue')
```

46211.325117016946

I have played around with the data. Above we could see how the model performs with data without manipulation. Actually, it is not needed because fulfill the stationary conditions required but, we have seen before that transforming the data using Log or Sqrt are much closer as Norm Distribution.

Either Log and Sqrt RMSE were much stronger than data without transformation showed much stronger RMSE, however, when we use the data for validate the model RMSE for Log data is much worse than data without transformation but Sqrt is slightly better.

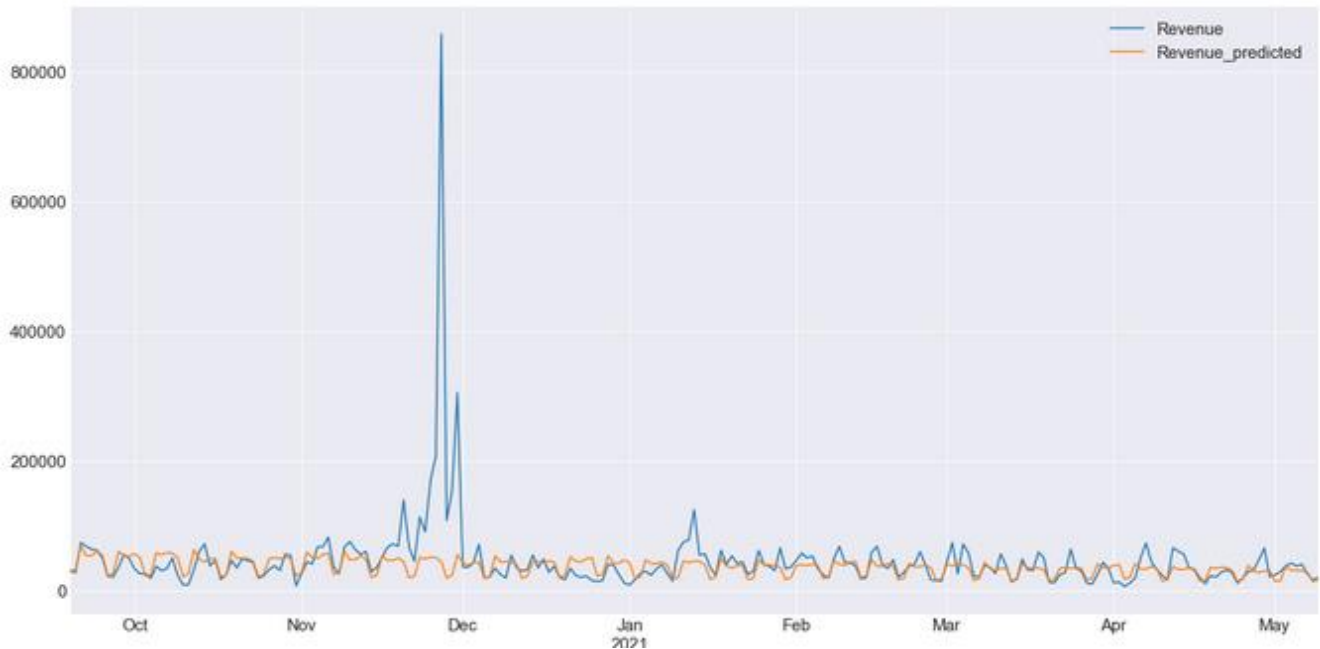
	Raw Data	Log Data	Sqrt Data
RMSE Train	22941	15415	19621
RMSE Test/Validation	46211	55824	45356

Triple Exponential Smoothing

This is another Classical Method and we applied all learnt in the SARIMAX model. Their results are worse than SARIMAX basically because it does not have the flexibility as SARIMAX for catching the seasonality and exogenous features.

RMSE Train: 26318

RMSE Test/Validation: 60830



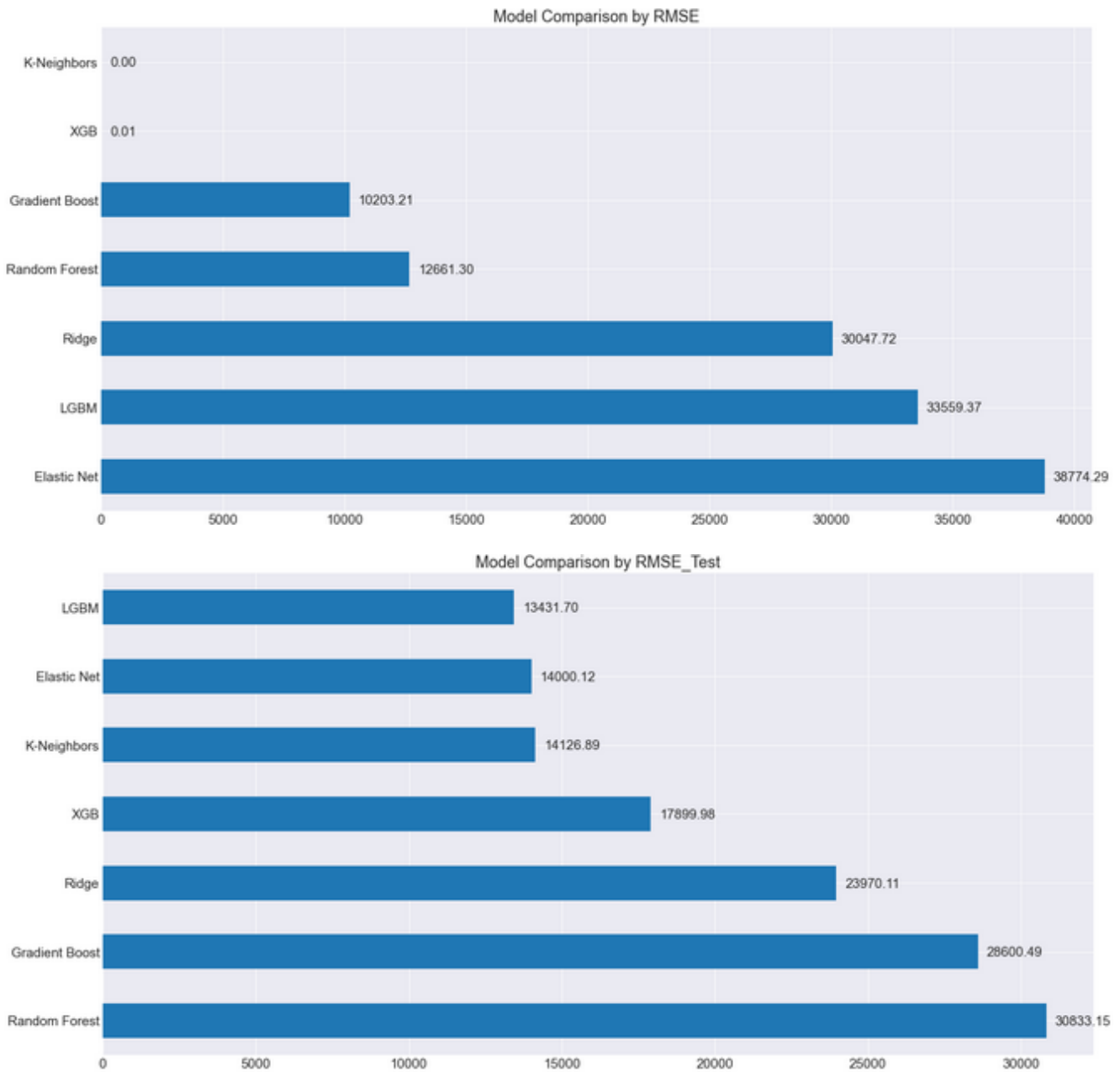
Machine Learnings Methods

This part was a challenge to me again for multiple reasons:

- Lag Target: Despite there is a method to do this automatically, need to understand the rational behind and use it wisely
- Seasonal features: Having choose these variables as categorial was an important decision which turned out good results
- Train Test Split: There is not any method which can do this task. I had to do it from scratch again as I mentioned at the beginning of this document
- Grid Search: As above, this method is already available for regular Machine Learning problems, but it is not valid for Time Series

In the notebook, you can find the different hyperparameters that we found for the different model used. This is an intensive task but way less than SARIMAX/TES Grid Search.

If we take a look how the different models perform based on RMSE Train and RMSE Test/Validation



As we can see there are two main models which are overfitting strongly, K-Neighbors and XGB. They are showing strong performance with the Train data and generalization is not so bad.

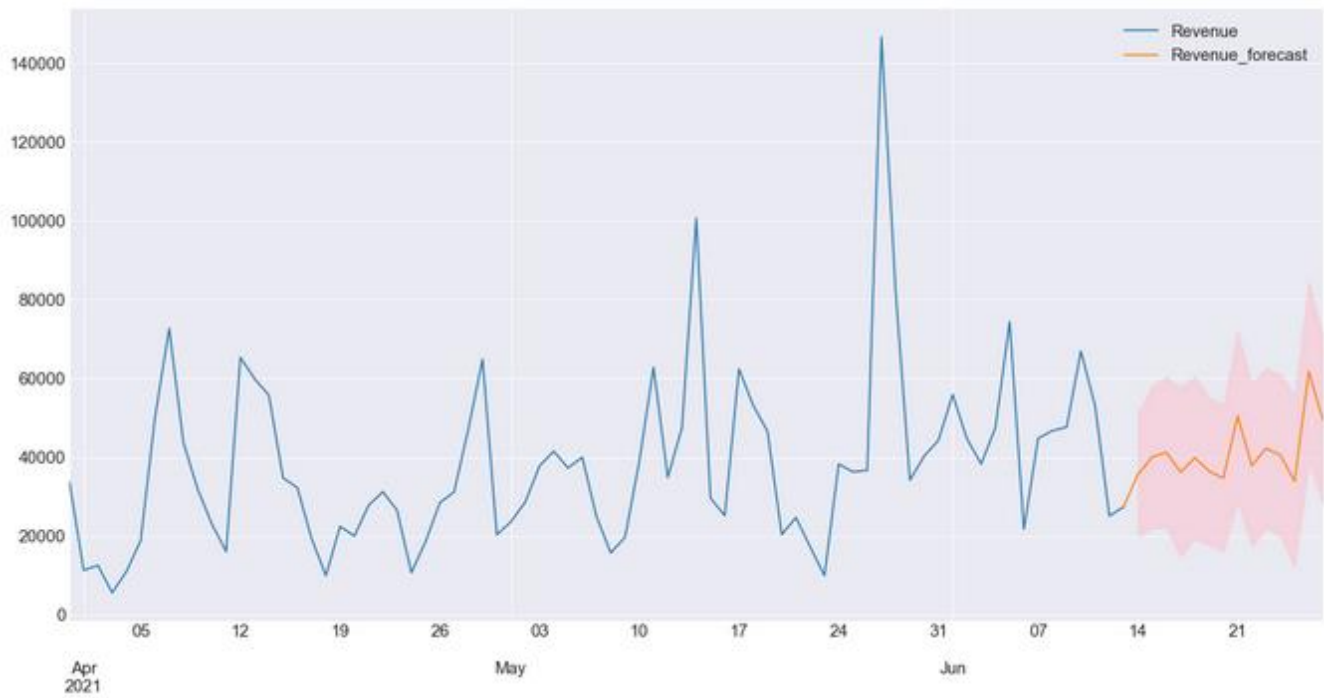
Forecasting

Now, let's see how the forecast part is performing. I have choose two of above Machine Learning Methods:

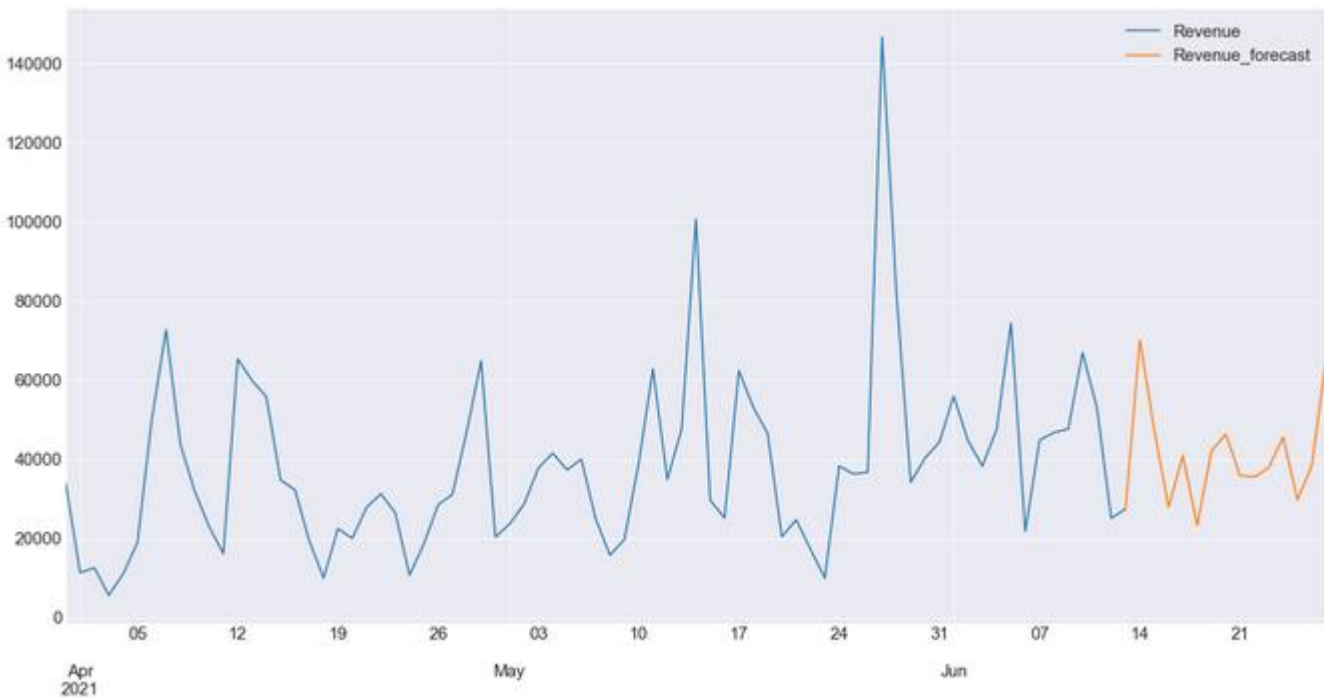
- Gradient Boost
- XGB

All those two models are forecasting 14 days and we are using 95% predictive intervals. The main challenge for this part was creating specific models for $T+1$, $T+2$... $T+n$. We cannot use $T+1$ for forecasting the $T+2$, so we have created a new method which create a specific DataFrame with all information. And train them separately with the hyperparameters that we found previously.

Gradient Boost Forecasting



XGB Forecasting



Alternative Methods

Just a briefly check with other methods, I have chosen another variation from a Classical Method, Prophet designed from Facebook and RNN using 5 layers, three of them hidden ones.

	Prophet	RNN
RMSE Train	31429	23171
RMSE Test/Validation	52114	32797

Conclusion and Next Steps

We have seen different models to resolve one important topic... forecast sales. This is more related with crystal ball skills than Data Scientist but we could see good results in some models.

An important learning that we need to consider for enhancing this analysis, could be taking the outliers off from the general model and create a specific model for outliers. In this case, Black Friday has affected and it will affect the predictions because it is a super abnormal value.

Next Steps:

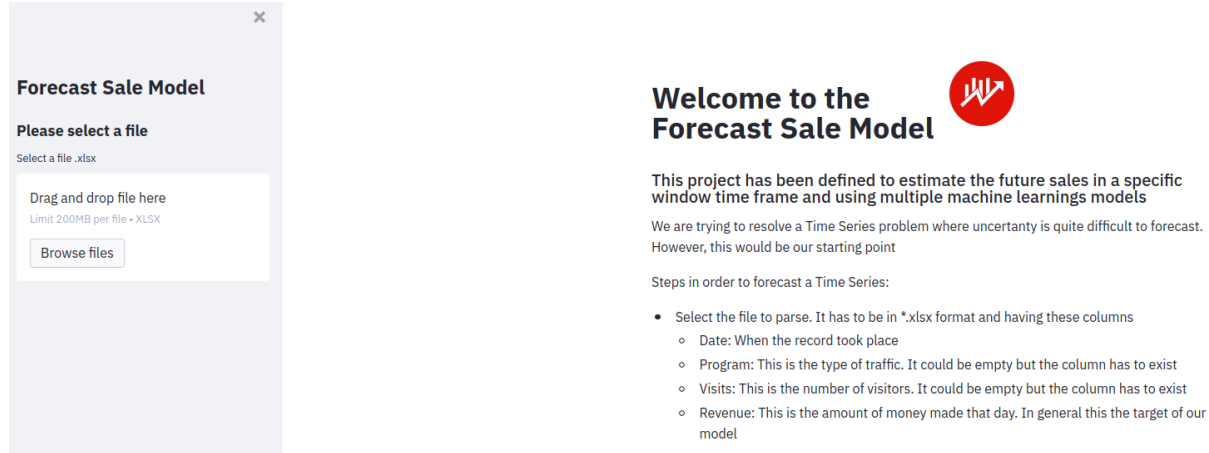
- Use Finance Data(100% confirmed) rather than Web Tracking Data(Less than 100% trusted)
- Create a separated model for outliers
- Add more exogenous features like weekdays, weekends, Christmas and bank holidays
- Add another lagged feature as Traffic and see how this performs
- Does it make sense to add stock level and lag this feature as well?
- Extending this analysis with more Deep Learning methods

Frontend

I have created a frontend for exploding what we have seen in this document. It has been developed in Python but it is using streamlit. You should type:

```
streamlit run visualization.py
```

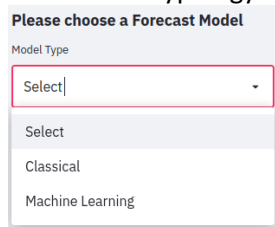
This is how this looks like:



The screenshot shows a web application titled "Forecast Sale Model". On the left, there is a sidebar with the title "Forecast Sale Model" and a section "Please select a file". Below this, it says "Select a file .xlsx" and "Drag and drop file here" with a note "Limit 200MB per file • XLSX". There is a "Browse files" button. On the right, there is a main content area with a red circular logo containing a white line graph. The title "Welcome to the Forecast Sale Model" is displayed. Below the title, there is a paragraph: "This project has been defined to estimate the future sales in a specific window time frame and using multiple machine learnings models". Another paragraph follows: "We are trying to resolve a Time Series problem where uncertainty is quite difficult to forecast. However, this would be our starting point". Below this, it says "Steps in order to forecast a Time Series:" followed by a bulleted list: "• Select the file to parse. It has to be in *.xlsx format and having these columns" with sub-points: "◦ Date: When the record took place", "◦ Program: This is the type of traffic. It could be empty but the column has to exist", "◦ Visits: This is the number of visitors. It could be empty but the column has to exist", and "◦ Revenue: This is the amount of money made that day. In general this the target of our model".

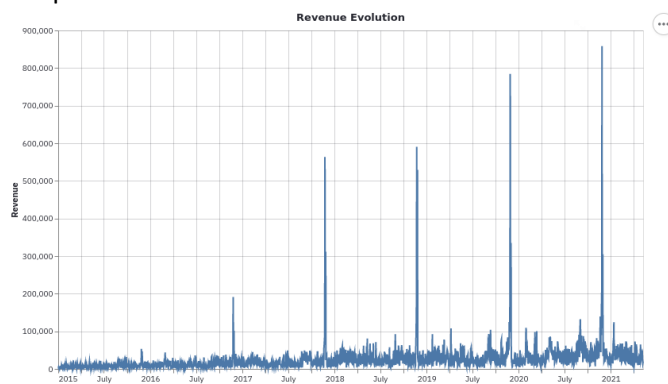
Steps:

1. Select a data source. We should pick data_v1.xlsx – This will take some time to execute it
2. Select what typology of method



The screenshot shows a dialog box titled "Please choose a Forecast Model". It has a "Model Type" label and a dropdown menu. The dropdown menu is open, showing the word "Select" at the top, followed by "Classical" and "Machine Learning".

3. Let's pick Classical. You will see all Dataset



- Choose the sub-model. When you make this selection, the program will train the forecast using the dates for training, you can choose the period. By default is one year starting from the beginning

Classical Type

SARIMAX

Select

SARIMAX

Holt-Winters

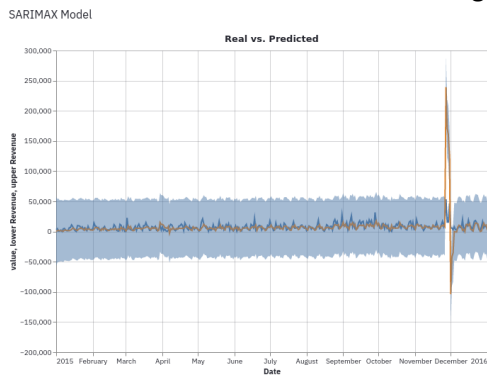
Chose starting date for training

2015/01/01

 Chose ending date for training

2016/01/01

- By default, we could forecast 2 days but we can increase this timeframe up to 28 days. Bear in mind, that the forecast will use the last training date



- In case you pick a Machine Learning. It is slightly different. Once you select one of the available methods, the Dataset will be trained online with the full data set using the hyperparameters of the Notebook. Bear in mind that this option is using one fold of the K-fold when we are doing Time Series Split and you could get different results because the model used has used just one random fold

Model Type

Machine Learning

Machine Learning Type

Select

Select

Ridge

Gradient Boost

XGB

Please choose a data range for prediction
 Chose starting date

2021/05/09

 How many days?

14

