Retail Forecasting

Clony Abreu
Pedro Gusmão

Prof. José Moreira
Profª. Ana Tomé

Types of Forecasting

- There are two types (quantitative vs qualitative)
- Studies show that using both can help in generating good forecasts (they can adjust results obtained by statistic methods or both can be combined after independent forecasts)
- If there is no data, qualitative methods are the choice
  - Delphi method, by analogy, etc...
  - Problems:
    - dependent on human cognition, memory etc...
    - can change with psychological factors

Focus: Quantitative Forecasting

- Assuming there is sufficient numerical data another condition must be met:
  - Believing that past structures are likely to happen in the future (data with seasonality)
- There are many options for quantitative methods and they are used in many fields
- Most quantitative prediction problems utilize time series data (collected over regular time intervals)

Statistical
vs
Machine Learning

Forecasting with Prophet

Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data.

PROPHET

Exploratory Data Analysis

The database content represents informations about 2 market stores named as store 432 and store 274. 51.3% of the data were found in store 274

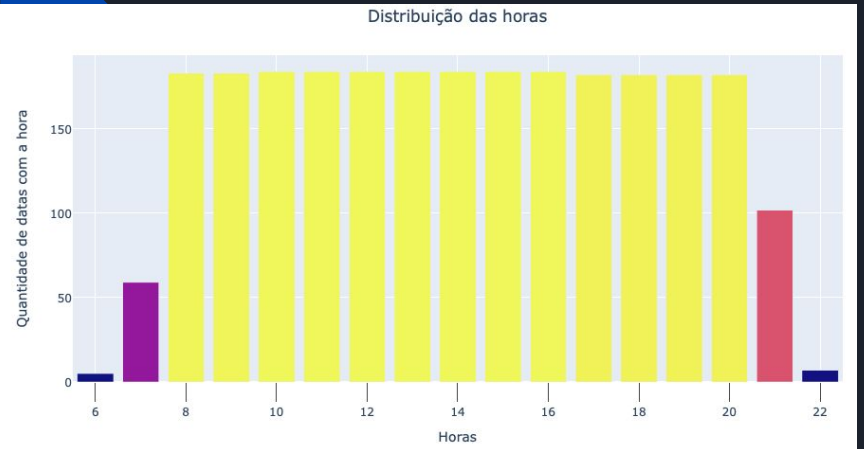
Exploratory Data Analysis

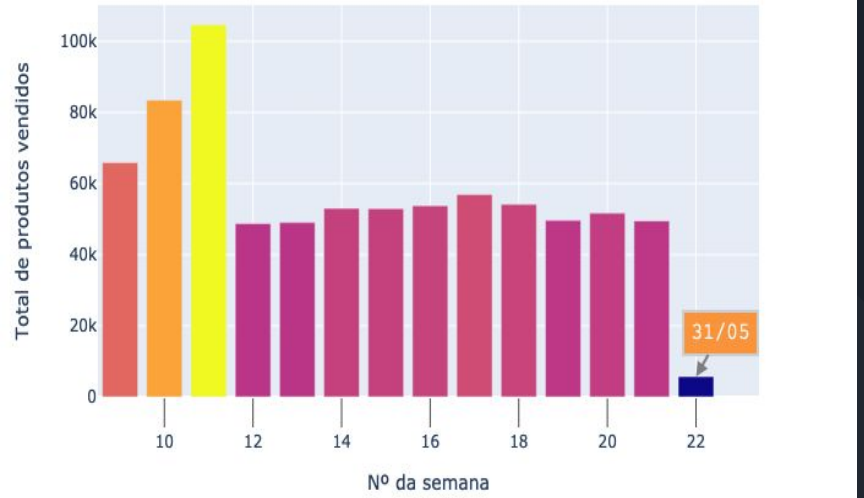
Negative values were found in the database, 100% of them in store 432

Exploratory Data Analysis

```
Data início: 2020-03-01 07:30:00, Data final: 2020-05-31 20:30:00
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2555 entries, 0 to 2554
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   ds      2555 non-null   datetime64[ns]
 1   y       2555 non-null   int64
dtypes: datetime64[ns](1), int64(1)
memory usage: 40.0 KB
```

Store 274 didn't had null values and the start date was at 07:30 am of 2020-03-01. The end date was at 20:30 of 2020-05-31.

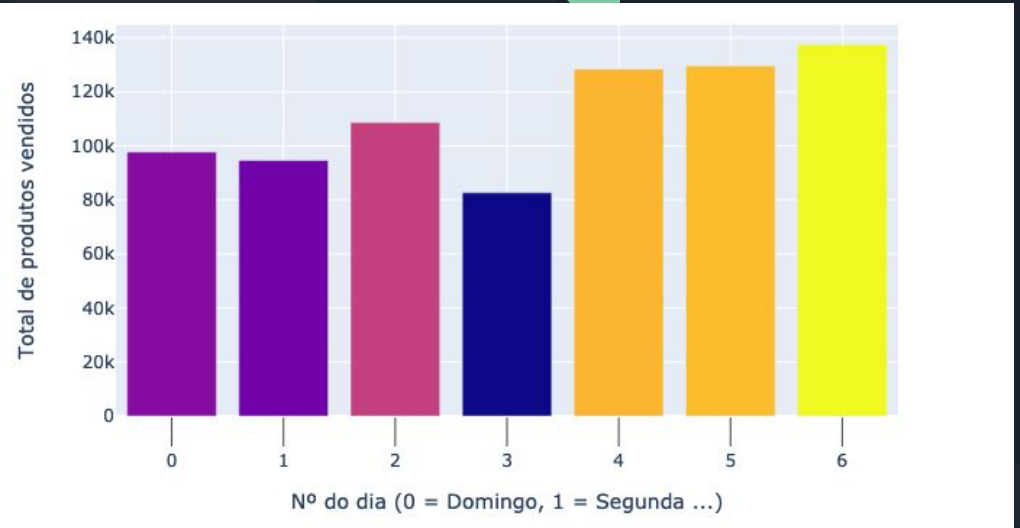




Exploratory Data Analysis

As store 274 has presented a more confident data, we started to look for the store opening and closing time. As the figure shows, there were not a fixed time for open and close. Then, for hourly analysis, the dataset was filtered to use records from 08:00 to 21:00. Also, the 22nd week is represented just by one day.

Exploratory Data Analysis

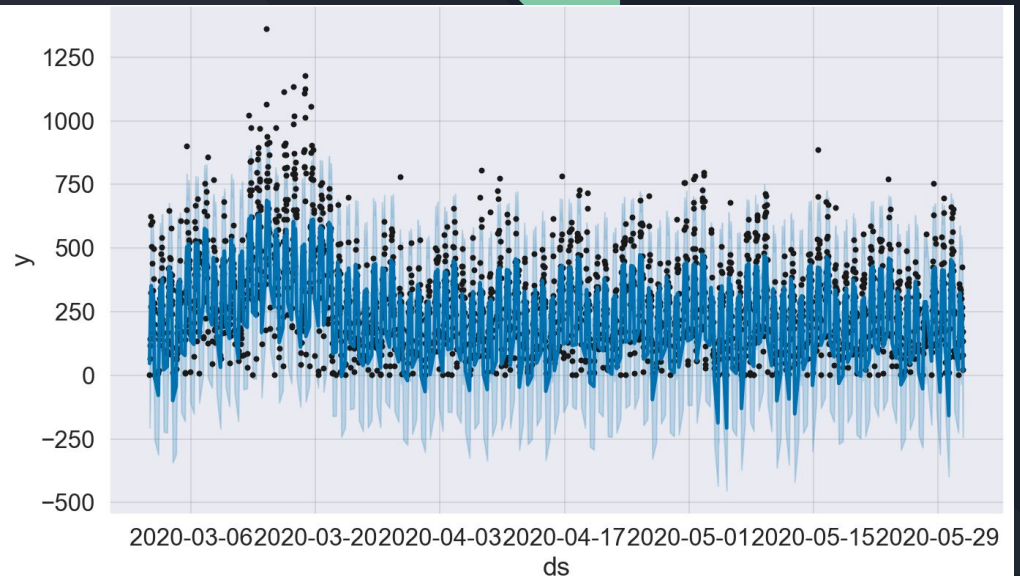


Wednesday is the one which present the lowest volume of products sell, by other hand, from friday to saturday are the ones which presents the highest volume of product sells.

Exploratory Data Analysis

The study started by hourly frequency. The figure represents the product sells for store 274 by hour. It is possible to realize a huge amount of products sell around march 12 to march 20.

Exploratory Data Analysis



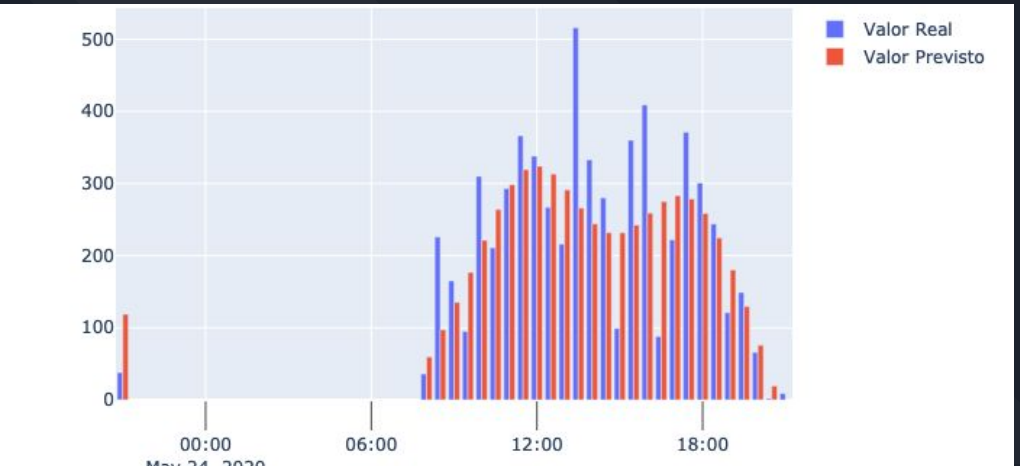
The first forecasting, without any consideration, just evidence some outliers and the prediction confidence interval, reveals some negative values. This might be happening due some data inconsistency or this time series has some particularity behave which is not reflected on data.

Exploratory Data Analysis


```
MAPE ------> 0.7240836968973005
```

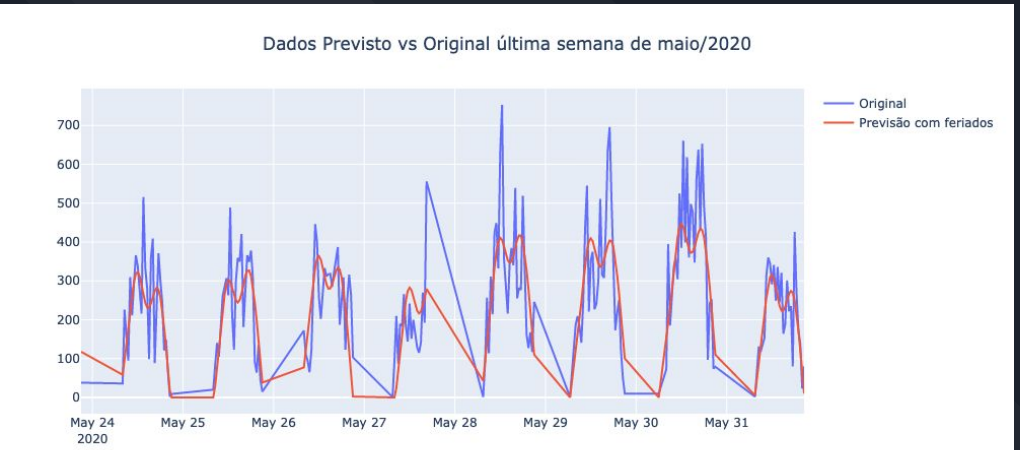
After preparing the train and test data set, and remove the negative values which we consider as a distortion of the reality due the low volume of data that might not being able to represents the seasonality and others particularity, we make a prediction for the last week of may and the MAPE result was of 0.72 of the real values.

Exploratory Data Analysis



Comparing the results we can see there is some outliers on may 24th from the test dataset, specially at 08:30, 13:30, 15:00, 15:30, 16:00 and 16:30. These outliers may represent some characteristics which the model could not predict due the low volume of data and may be compromising the model performance.

Exploratory Data Analysis

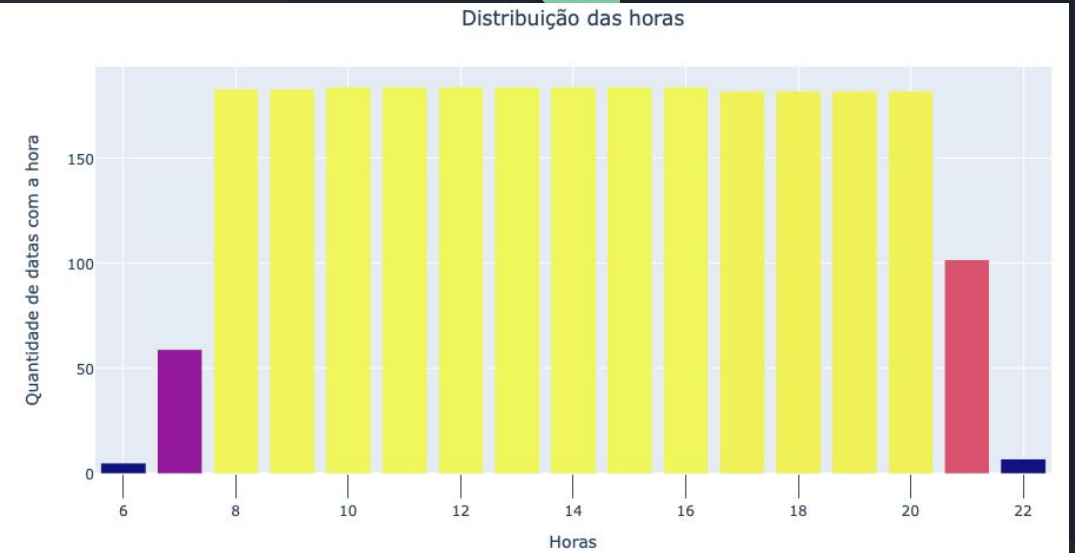


We know that the time series has a seasonality but the amount of records is not enough to make Prophet predict with higher accuracy, even adding holydays the model improves a bit in its performance getting a MAPE of 0.71, which still very high.

Exploratory Data Analysis

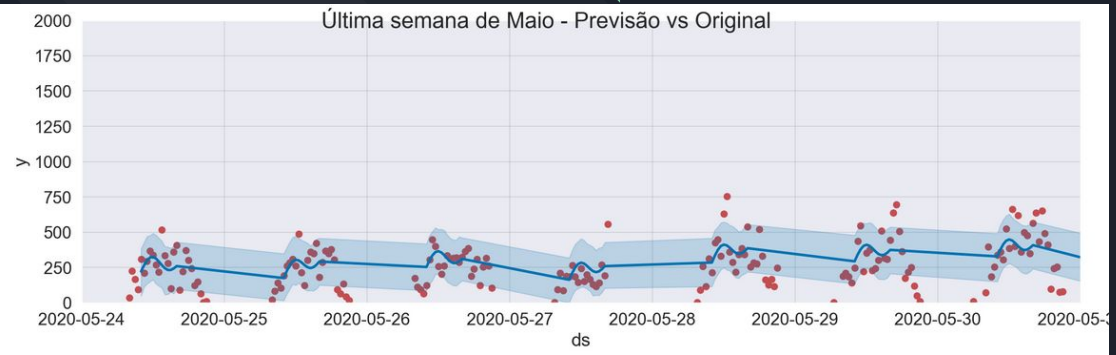
Monday seems to be the worst selling day according to this plot, but we know it is wednesday, although, Prophets can deal with seasonality very well, it is possible to correct this behaviour by adding some regressors, but it is necessary to go deeper with business experts.

Exploratory Data Analysis



Prophets can deal with regular gaps and sub-daily data, such as hourly our half-hour data. As we've seen before, there is lots of gaps on opening hour but, the time between 10:00 and 16:00 has the most amount of data. Prophets works very well with missing values, and for that it is enough to configure the parameters correctly. So we filter the dataset just for this period (10 - 16) and try again to predict.

Exploratory Data Analysis




```
print("MAPE ------> {}".format(fm.mape(np.array(df_t_new['y']),
                                       np.array(df_274_sale_teste_forecast['yhat']))))
```

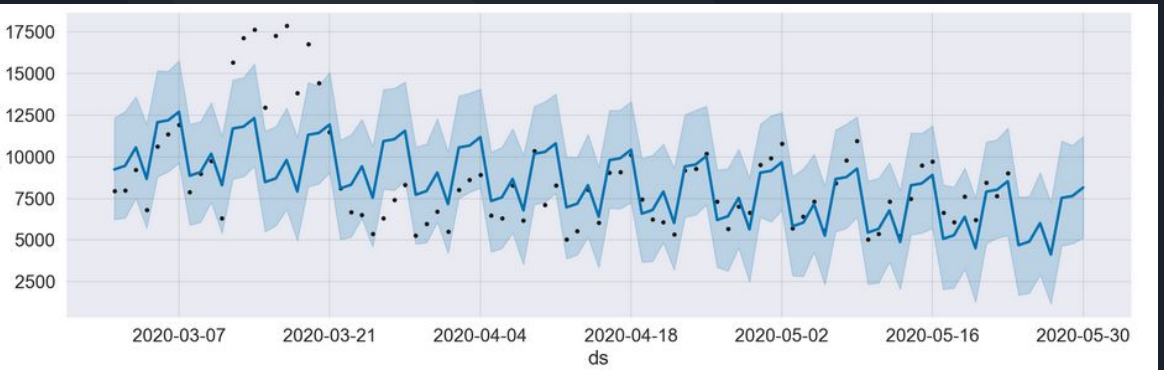
```
MAPE ------> 0.28547739903285607
```

We can see that the predicted and real values are closer to each other in this visualization and the MAPE confirms the improvement with a 0.285 as result. this gives us a clue on how to handle hourly distributed data in the future.

Exploratory Data Analysis

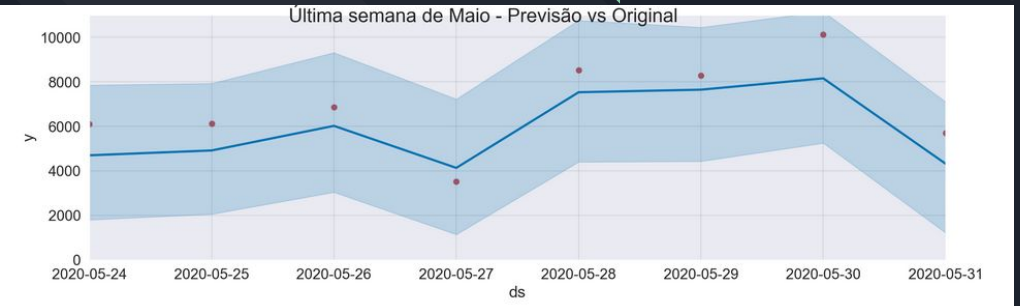
In order to deal with the seasonality problem observed with hourly dataset, we use the 274 store dataset, grouped by day. The overall aspect of the dataset is very similar to the hourly one.

Exploratory Data Analysis



In this scenario the forecast trend has a descending aspect. The black dots represents the real records, the blue line is the forecast and the light blue area is the confidence interval.

Exploratory Data Analysis

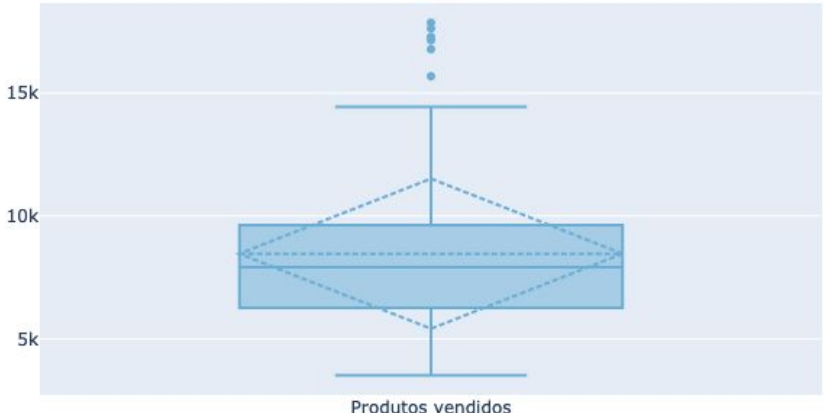



```
mape_analysis = []
mape_analysis.append(fm.mape(np.array(df_274_sale_teste_diario['y']),
                             np.array(df_274_sale_teste_diario_forecast['yhat'])))
print("MAPE ------> {}".format(mape_analysis[0]))
```

```
MAPE ------> 0.16899542665715367
```

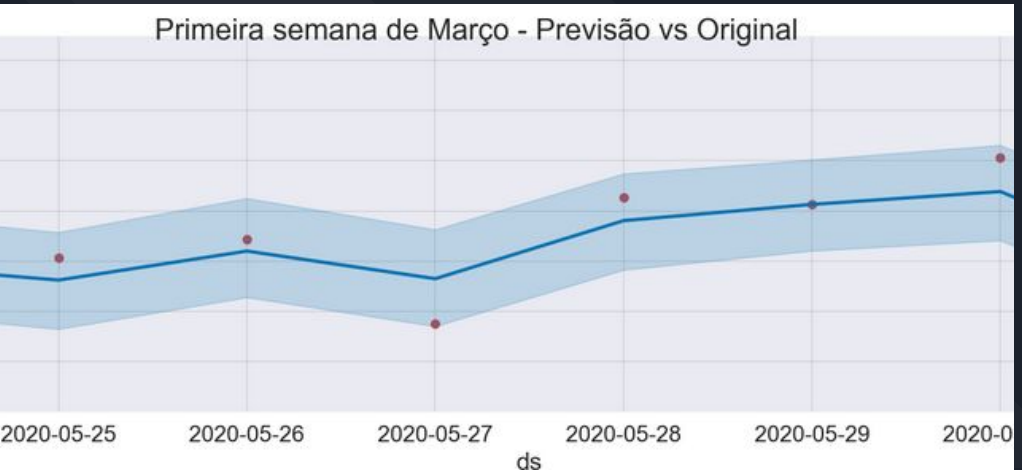
In this first prediction we can see that the forecast has a better result. The red dots (real records) are very close to the blue line (predicted records) and it is inside the light blue area (confidence interval), this behave represents a MAPE of 0.168 .

Exploratory Data Analysis



Although outliers may be a result of an intentional action and sometimes it is desired, we don't have this information so, we use a very simple way to remove outliers safely without cause any overfitting or underfitting. We use a boxplot to identify the records which are not consider the expected (the ones inside the box), so, those data outside the upper fence are considered to be outliers.

Exploratory Data Analysis

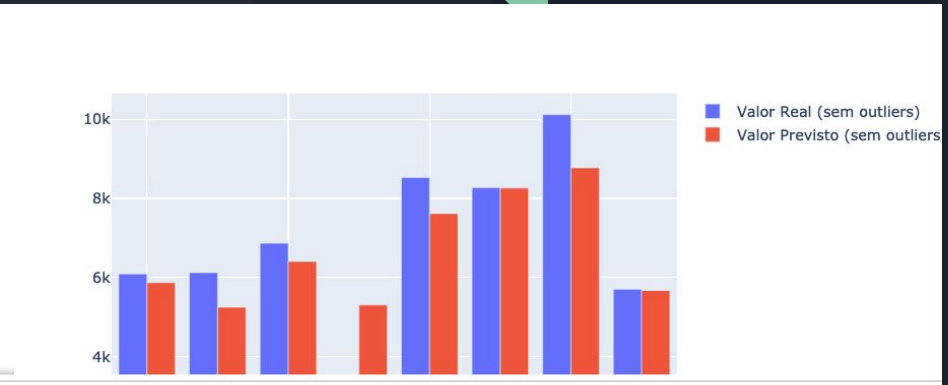


The forecasts without the outliers is more accurate as reveals the figure.

Exploratory Data Analysis

Visualizing with bar plots gives a good idea how close the prediction gets, but still some outliers can be seen on may 27th. In this particular case we found that the store records are finished at 16:30, so the dataset is missing in 4 and a half hours of data for this day, and that's why it looks like an outlier but we can't confirm this.

Exploratory Data Analysis



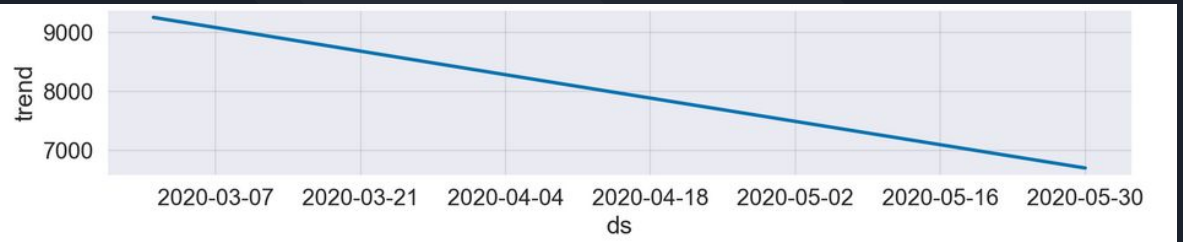

```
n [61]: mape_analysis.append(fm.mape(np.array(df_274_sem_outliers_teste.y),
                                     np.array(df_274_no_outliers_teste_forecast.yhat)))
        print("MAPE -----> {}".format(mape_analysis[3]))

        MAPE -----> 0.12482887010004967
```

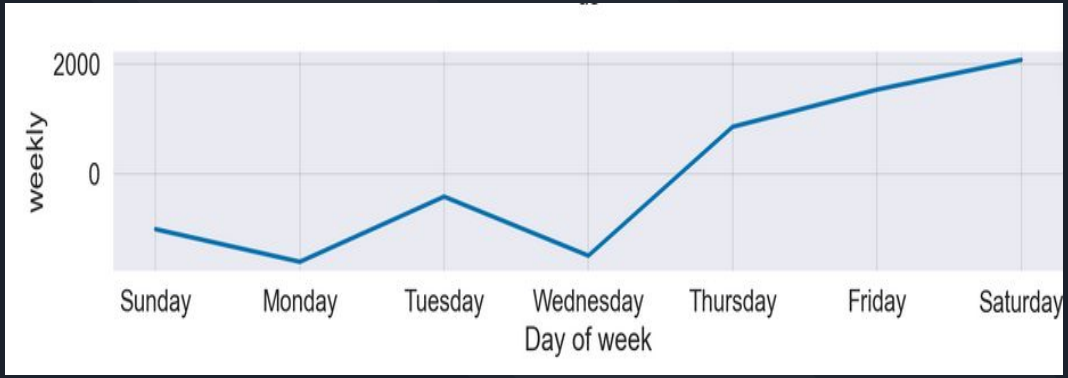
As we could expect, the MAPE (0,1248) was improved in almost 5%.

Exploratory Data Analysis

As the dataset presents a very huge amount of products sells on march and "never" reaches these values again, the forecast has a down trend.



Exploratory Data Analysis

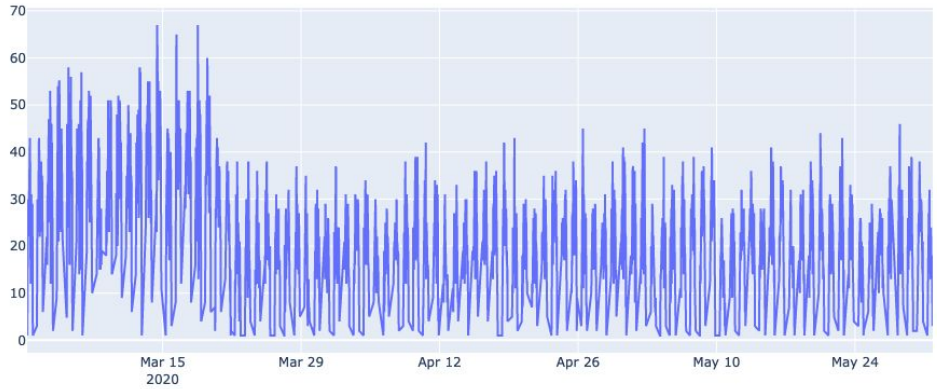


We know that monday is not a very good selling performing day, but the data shows that wednesday is worse. Removing outliers, makes a correction on this seasonality.

Exploratory Data Analysis

Number of clients

Exploratory Data Analysis

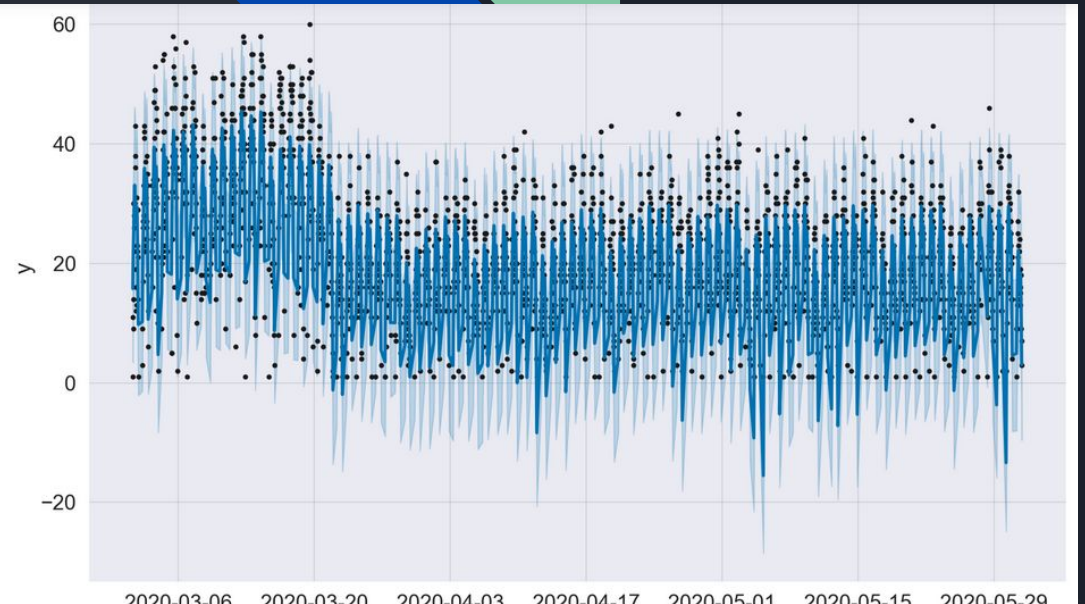


The number of clients on store 274 has a similar shape with the sells time series previously studied.

Exploratory Data Analysis

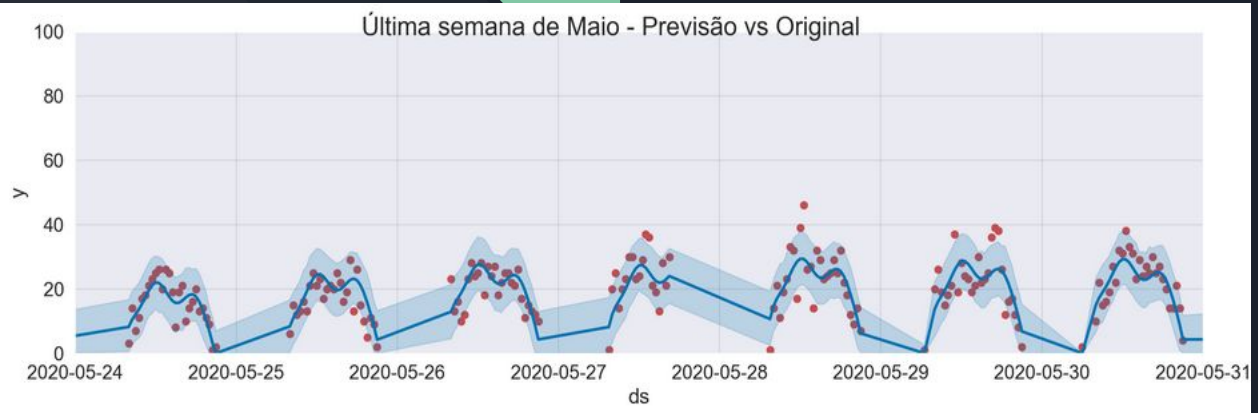
When analysing per day we can see a difference from the first time serie. The number of clients increase as the days goes by until thursday (which is the peak) and keep higher then the other days. Sunday is the day with the lowest number of clients.

Exploratory Data Analysis



The first forecast reveals several negative predictions

Exploratory Data Analysis

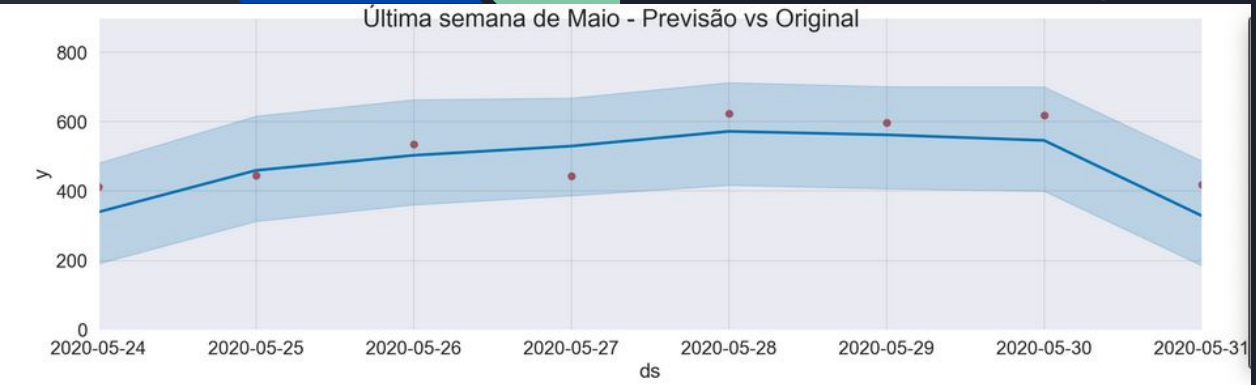



```
print("MAPE ------> {}".format(fm.mape(np.array(df_274_clients_teste['y']),
                                       np.array(df_274_clients_teste_forecast['y
```

```
MAPE ------> 0.36810083250277376
```

The forecast has a good initial MAPE value (0.37), but the same issues seeing on the sells time series, impacts on the results and it didn't get any better.

Exploratory Data Analysis

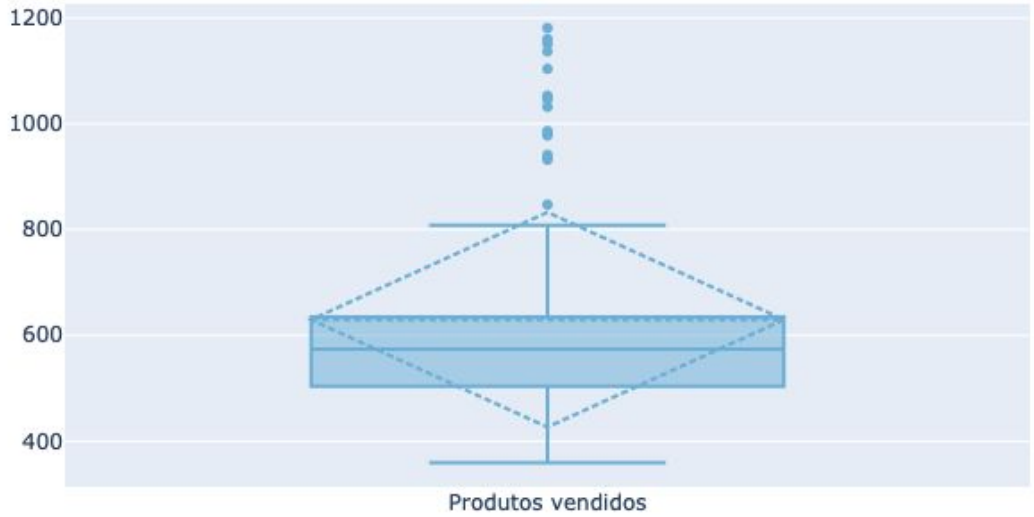



```
mape_analysis = []
mape_analysis.append(fm.mape(np.array(df_274_clients_teste_diario['y']),
                             np.array(df_274_clients_teste_diario_forecast['yhat'])))
print("MAPE ------> {}".format(mape_analysis[0]))
```

```
MAPE ------> 0.11685400904728263
```

Grouping by day we can see the red dots (real values) get closer to the prediction (blue solid line) and inside the confidence interval (light blue area),which lead to a 0.11 of MAPE result.

Exploratory Data Analysis

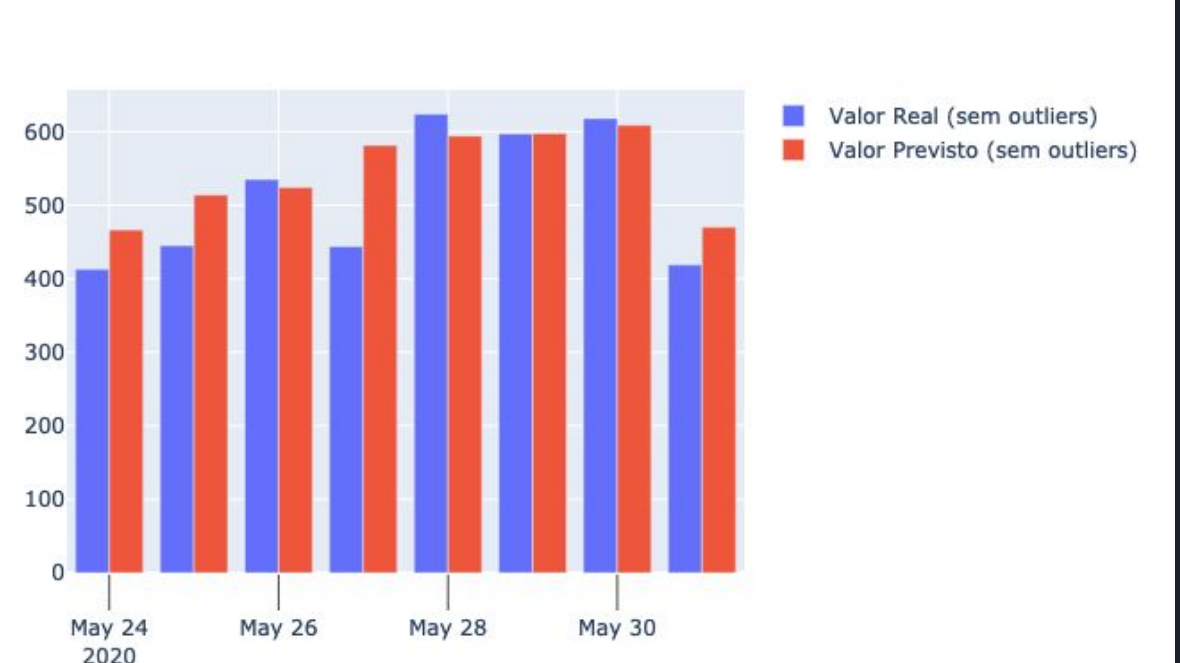


Grouping by day and removing outliers with boxplot

Exploratory Data Analysis

The unexpected form of the test dataset is due the may 27th already mentioned problem, that the store does not have data after 16:30 so, it obviously has less clients. The curve contradicts the expected behave hence may 27th was a wednesday and should have had more client then tuesday.

Exploratory Data Analysis



Viewing as a bar plot it is possible to verify the problem on may 27th, still, the results are positives.

Exploratory Data Analysis

```
mape_analysis.append(fm.mape(np.array(df_274_sem_outliers_teste.y),
                             np.array(df_274_no_outliers_teste_forecast.yhat)))
print("MAPE -----> {}".format(mape_analysis[2]))
```

```
MAPE -----> 0.10003574900130739
```

- Prophets achieve a good result predicting the number of clients removing outliers and grouping data by day.

Prophets Benefits

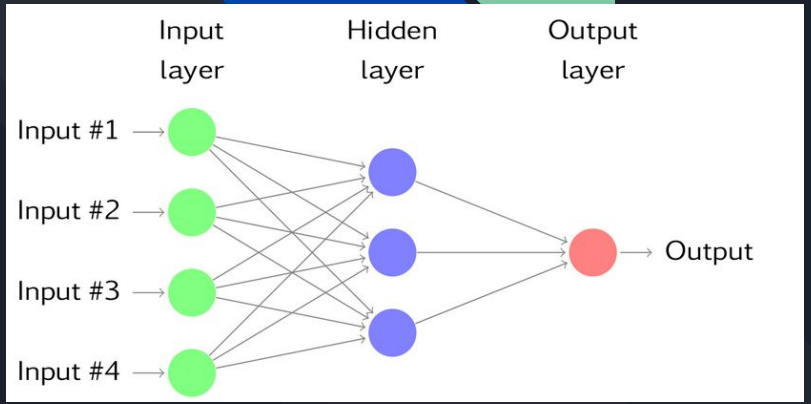
Prophets, when correctly used and adapted to a time series with strong seasonality can achieve very good results with less time, less effort and withe lower computation resources, the main focus here is the business variables and characteristics.

PROPHET

Data Analysis & Business Doubts

1. What is the meaning of "*importe_efectivo*" column (can we assume it is like IVA or some kind of tax?);
2. Column "*valuepln*" has the same meaning of "*importe-efectivo*"?;
3. Why exists negative values for *valuepln (importe_efectivo),* products (*units*) and clients columns?
    a. Is this an error or there is a meaning for that?
4. Why there exists 801 values with different closing time on the new dataset for store 387?
    a. Are these cases only for sundays?
        i. If so, why some sundays last datetime data are at different time?
5. Why there exists different opening time on the new dataset for store 387?
6. Some records has outliers values on closing time (2020-06-06 22:00-22:30) for units and clients, what they represent for the business (was a campaign?)?
7. Is there any mapped knowledge for seasonality?

Neural Networks



A neural network with four inputs and one hidden layer with three hidden neurons.

Can be seen as a network of neurons organized by layers.

- In the context of time series the output represents the forecast while the inputs are the predictors (e.g. lagged observations for autoregressive models)
- The existence of hidden layers is what makes possible for the network to discover complex nonlinear relationships between the predictors and the target variable (activation functions), otherwise it would be similar to a linear regression

NNs vs Classical Statistical models

- As mentioned, NNs can capture nonlinear relationships
- Are much more flexible and configurable models
- Are a better fit for multivariate models, there are few statistical models for this

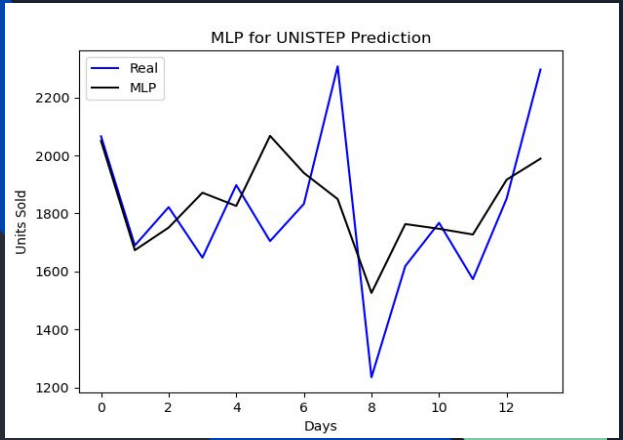
- The need of training → very data **hungry**, can take a long time to process
- Some architectures are very complex → a wide range of parameters to tune
- Task of data preprocessing is usually harder

- But both can be used together to gather interesting results → M4 Competition:
  - Exponential Smoothing + Recurrent Neural Networks (ESRNN)

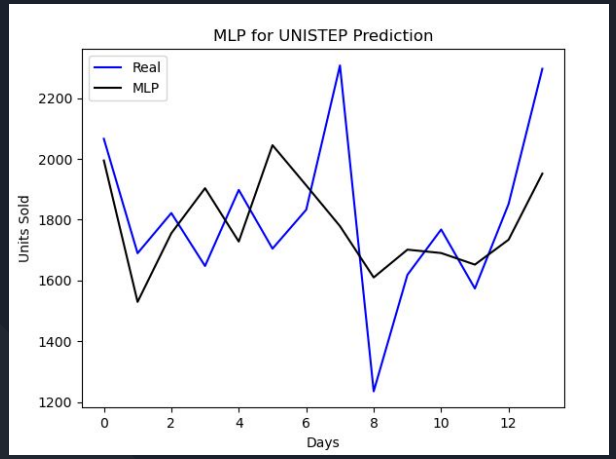
DISCLAIMER!

- The following models were not tuned
- They only utilize weekday and units sold as variables
- Lower errors with tickets

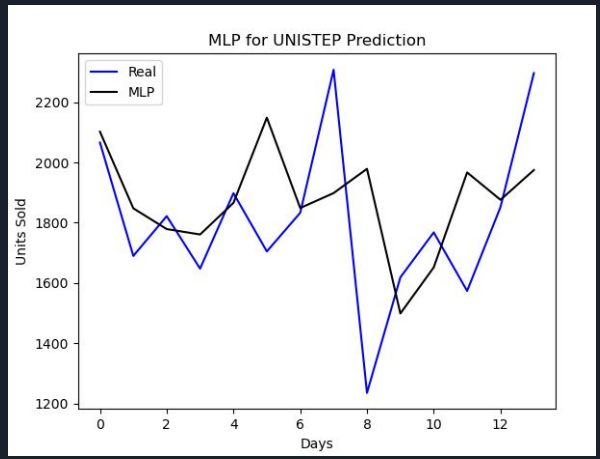
One Step at a Time: UNISTEP Predictions



Lagged observations = 14
MAPE = 0.093
RMSE = 213.898



Lagged observations = 7
MAPE = 0.108
RMSE = 242.176



Lagged observations = 1
MAPE = 0.129
RMSE = 297.989

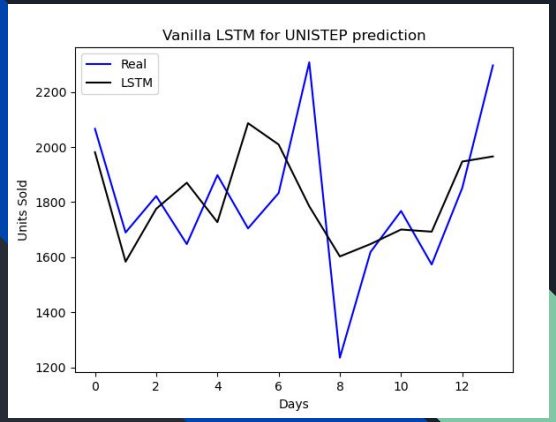
MLP w/ Daily data

The features used for every approach are lagged observations (NNAR) and day of the week.

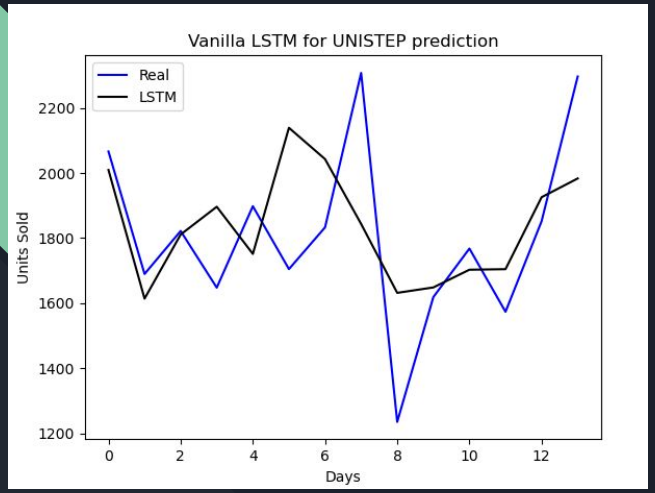
It's no use predicting 30 minutes ahead once at a time.

Store 1027 was used. Last 2 weeks predicted, one day at a time.

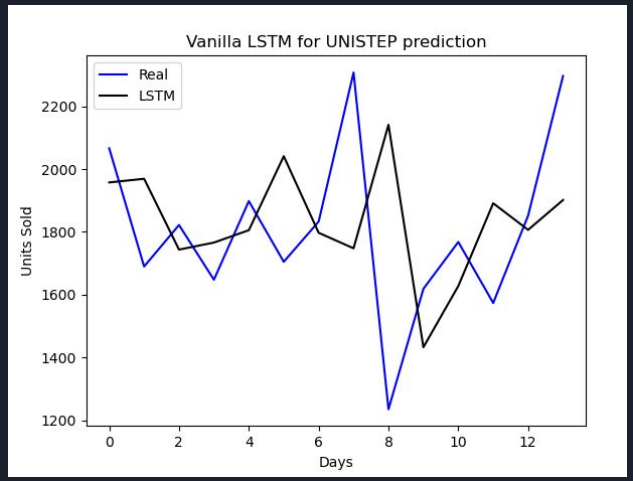
One Step at a Time: UNISTEP Predictions



Lagged observations = 14
MAPE = 0.109
RMSE = 242.065



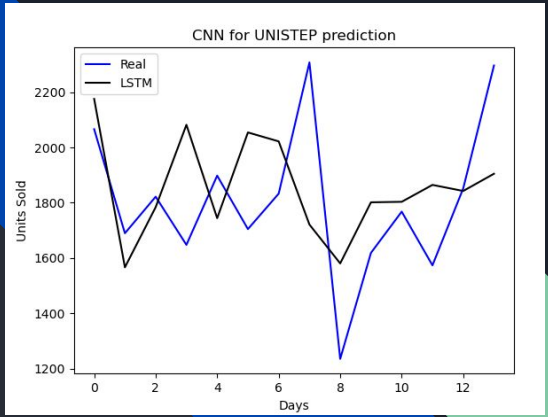
Lagged observations = 7
MAPE = 0.108
RMSE = 242.219



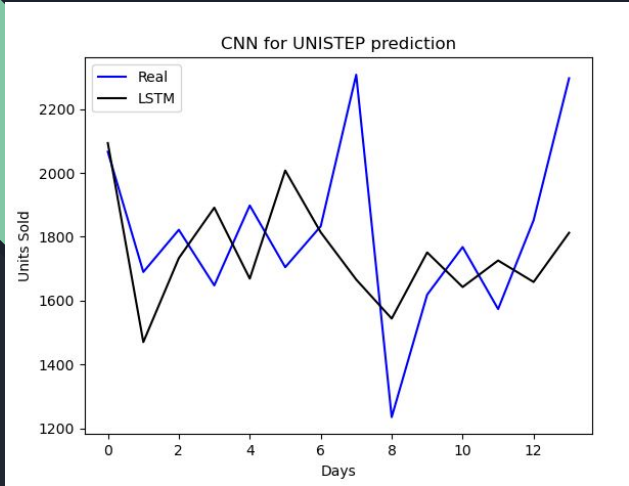
Lagged observations = 1
MAPE = 0.155
RMSE = 346.002

LSTM w/ Daily Data

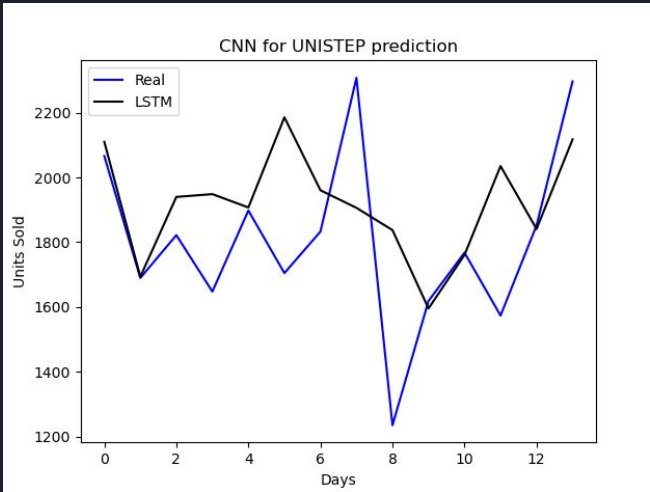
One Step at a Time: UNISTEP Predictions



Lagged observations = 14
MAPE = 0.130
RMSE = 284.355



Lagged observations = 7
MAPE = 0.124
RMSE = 279.372

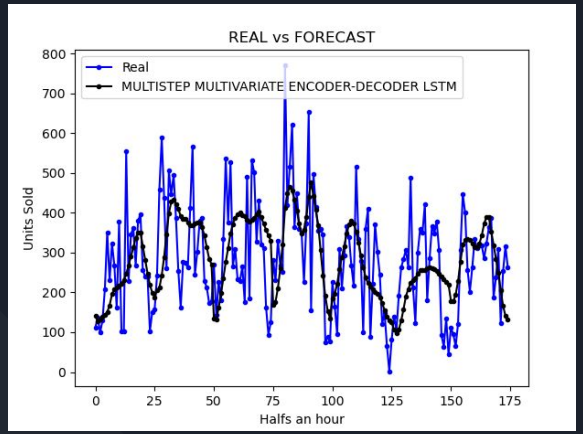


Lagged observations = 1
MAPE = 0.120
RMSE = 282.917

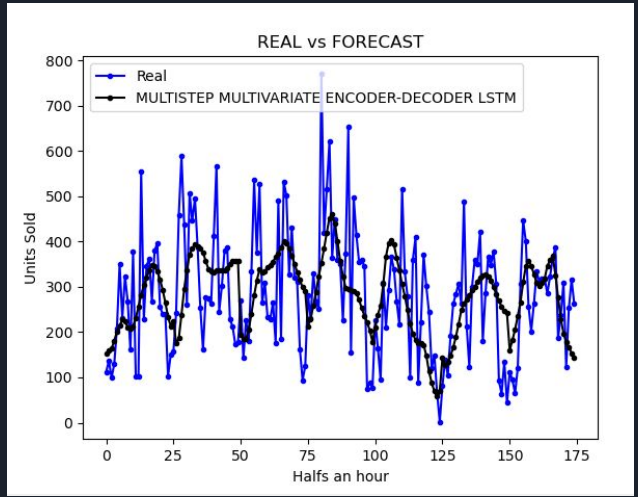
CNN w/ Daily Data

Multiple Steps at a Time: MULTISTEP Predictions

Lagged observations = 75



Lagged observations = 50



Lagged observations = 25

Encoder Decoder LSTM w/ Half Hourly Data

Store 247 was used.

Time intervals were sliced from 08:30 till 20:30.

The predicted steps are 25 at a time. (An entire day for seven days = 175 steps)

A few gaps exist in the data, they were handled in an attempt to not disrupt the prediction model.