Note -The following document is for “Technical case study” in the Assignment Provided.

Table of Contents

Approach to the problem

Understanding the data

The data contains 197 different series for 197 Products. All individual Products will have their own model. Within the model, we will predict Quantity of Product sold every day. We can use Promotion data and Product Retail Cost as additional data to model to further improve the model.

We would also need to calculate URP after applying Promotions. Have created the method “retail_price_after_promotion” to update URP.

Cleaning the data

Firstly, we tried to understand that which Products are being actually sold and which seems to be dis-continued. We made the assumption that any Product which didn’t had any sales from last 6 months was obsolete or any Product which had < 90 days of overall data didn’t had enough information to create a model from.

If we are implementing this solution on Production, then for “obsolete” or Products with less data, we can create a simple Moving Averages model to ensure that there is a completeness of Product in our model. This does needs to be communicated to client.

Imputing the data

Most Time Series Algorithms cannot work effectively with missing data. Therefore, missing date values needs to be imputed. Here, we need to ensure that imputation is done for individual products separately.

I have used simple forward-fill method to impute missing values. We can also use nearest neighbour to impute any missing data.

I have created the method “date_impute” to impute missing dates.

Exploring models

First, we would ideally create a simple ARIMA or SARIMA model to make a baseline. For this, we would first need to remove trend and seasonality like? from time-series data to make data stationary.

Using Cross-validation for Time series will ensure that model is generalized and not overfitting.

Models can be evaluated based on MAPE or RMSE. I have used MAPE as it is a good way to evaluate time series model and its easier to understand by both non-technical and technical clients.

I have used FB Prophet model for simplicity in the solution. To make model more sensitive to trend and seasonality we could also do hyperparameter-tuning. Exogeneous variables can also be added for further improving the model.

Implementing the models

We would need to feed data in a loop to create individual model to every Product. So, number of time-series will be equal to number of products.

Production Movement

We can display the data on a visualization tool like Tableau or PowerBI which can be used for Budget planning.

Once all stakeholders are satisfied with the model, team can work with Data Engineers to Schedule and create pipelines to automatically update the model and push results on Tableau at predefined cadence.

Questions for Business / Data Understanding

1. Detect the outliers in the quantity and apply an outlier treatment on the same. Specify the outlier treatment technique and reason for choosing the same

I identified outliers based on the behaviour of products.

1. Any product which has not been sold for last 6 month, can be dropped from prediction as we are assuming that it's an obsolete product.

2. Any product where total days of sales is less than 90 as we would like to predict for next 90 days and having at least 1 cycle worth of actual data is necessary to create model. Also, to ensure that we are not missing any product that recently got active, we will only drop those products that appeared less than 20 times in last 6 months.

Here, ideally the aim is to 'save' as many products as possible for forecasting. We will only drop extreme values. Additionally, if this solution is put in production and any specific product reappears in data and satisfies above two conditions, then they won't be dropped.

2. Check the data for missing values and apply the missing value treatment. Also provide the reasoning for the missing value treatment method used

Most Time Series Algorithms cannot work effectively with missing data. Therefore, missing date values needs to be imputed. Here, we need to ensure that imputation is done for individual products separately.

I have used simple forward-fill method to impute missing values. We can also use nearest neighbour to impute any missing data. I have created the method "date_impute" to impute missing dates.

3. Check for seasonality and trend effect and smoothen if needed

In my solution, I have used FB Prophet which has an inbuild hyperparameters which can be tuned to take care of the impact of seasonality and trend. They can be made more or less flexible based on how well the model is performing on Validation set. We can use a grid search with various values of seasonality and trend flexibility to find out the best set of parameters.

In my current solution, I have not added Grid Search for simplicity.

It would be useful to use smoothening techniques if the data has outliers. Smoothening also helps to ensure that our predictions are not capturing one-off events in the time-series data. For this exercise, I have not smoothen the data.

4. Results of EDA performed, if any

```
agg_data.head()
✓ 0.6s
```

Product	min	max	count	date_diff	missing_data	missing_data_percent	count_in_last_6_months
Product 1	2017-01-01	2019-12-31	1086	1095	9	0.83	182.0
Product 10	2017-01-01	2019-12-31	1086	1095	9	0.83	182.0
Product 100	2017-01-01	2019-12-31	1086	1095	9	0.83	182.0
Product 101	2017-04-24	2019-12-31	974	982	8	0.82	182.0
Product 102	2017-04-26	2019-12-31	972	980	8	0.82	182.0

In EDA, I tried to understand the behaviour of every Product series. I created a table with above columns, which I used later to remove any outlier and impute data.

5. Fit a model to predict the quantity from 1st Jan 2020 to 31st Mar 2020

I have fit a FB Prophet model for the predictions. I have used FB Prophet model without hyper-parameter tuning for simplicity.

If we were building a production ready model, then would also need to do the following:

1. Check for stationarity of time-series.

2. Start with simpler models like Arima, Sarima before utilizing complex models like Prophet or LSTM

3. Check for bias variance tradeoff. One way we can check it is by comparing training and testing error. If testing error is too high as compared to training error, then there are chances of overfitting the model.

4. Here I have used a simple Prophet model. For such complex model, its worth using grid search to identify best performing hyper-parameters

5. We can also include exogeneous variables in the model like holiday data, promotion data etc.

6. We can also create clusters of Product based on behaviour.

6. Evaluate your model performance

Evaluating used MAPE.

After removing unusable Products, we created model for 134 Products. Following is the distribution of Test MAPE. We used last 6 months as Test Dataset. We have attached a PDF to visualize the performance of few top Products (ordered by total quantity sold)