Demand Forecasting

Prediction of upcoming sales is a part of planning to run a balanced business. It helps understand and be ready to serve the customers during unexpected demand and at the same time avoid inventory wastage during less demand days. Prediction of demand can be done with domain expertise, using moving average calculation. However, with the advent of AI, it provides an opportunity to make use of the past demand patterns and forecast the upcoming sales demand with more confidence and accuracy. This paper talks about developing a model based on machine learning, selecting the best model that fits a retail business and various techniques used to validate the model.

Retail client mentioned above has multiple vendors that serve various retailers in multiple routes. As of now, the client uses domain expertise to maintain inventory in all the retail stores. Data available in the system of the client provides scope to make use of the past data sales or demand patterns to predict the future demand.

Various features considered during the model development are:

* QOHCalculated – Inventory maintained in the shelf.
* Day\_of\_Week – Day name of the week.
* StoreNumber – Store number referred by the business.
* StoreID – Store number referred by IT / developers.
* storeCity – City where a retail store is located.
* storeState – State where a retail store is located.
* StoreZip5 – Zipcode where a retail store is located.
* ProductID – SKU ID
* ProductName – SKU Name
* RetailPrice – Sale price of SKU.
* ActualSaleDate – Sale date of the SKU.
* VendorId – ID of the Supplier who supplies the SKU.
* VendorName – Name of the Supplier who supplies the SKU.
* RetailerID – Owner ID of a store who sells a SKU to customer.
* Retailer – Owner name of a store who sells a SKU to customer.
* QtyDelivered – Quantity of SKUs delivered to a retailer by vendor.
* QtyReturned – Expired Quantity of SKUs returned to vendor by retailer.
* QtySold – Quantity of SKUs sold at a store.
* RouteID – Route a vendor choose to deliver a SKU to a store.
* Season – Winter/Spring/Summer Season in a year.
* Sales\_7\_Days\_Lag – Qty sold at store /Qty Sold in corresponding zip for a SKU 7 days back from current day.
* Sales\_14\_Days\_Lag – Qty sold at store /Qty Sold in corresponding zip for a SKU 14 days back from current day.
* Sales\_21\_Days\_Lag – Qty sold at store /Qty Sold in corresponding zip for a SKU 21 days back from current day.
* Sales\_28\_Days\_Lag – Qty sold at store /Qty Sold in corresponding zip for a SKU 28 days back from current day.
* Tavg – Average temperature at a city on a given day(F).
* Wspd – Windspeed at a city on a given day (Km/h).
* Prcp – Precipitation at a City on a given day (mm).
* Snow – Snowfall at a city on a given day (Inches).
* Holiday – Is it a US Public Holiday.

Few points to understand the data are:

* A single retailer can have more than 1 store.
* A single store can have more than 1 vendors to supply a same product.
* A single vendor can use multiple routes on different days to deliver SKU to same retailer.
* A single store can have different prices for same SKU on different days. Most of the days, a SKU would have same retail price.
* Weather data is extracted using the module metostat in python.

To develop a demand forecasting model, there are multiple approaches, calculations, and algorithms available. Some techniques that are attempted during development of the model are listed here:

1. Simple moving average model – take average of the past week/s sale/demand to predict the upcoming week sale/demand.
2. Time series model using FB Prophet that captures the seasonality of past sale/demand and predicts the upcoming sale/demand.
3. Regression model that takes inputs of various independent factors to predict sale/demand – used Multi linear regression, Decision tree, Random Forest and XG-Boost.

Of all the approaches, Random Forest model delivered best results.

Since the data comprises multiple SKUs and stores, If, a single model is used for prediction, there is a high chance of model getting biased results as some SKUs would be seasonal products and some stores would have high sales. Thus, one model for each store-SKU combination is developed. In this way, there would be multiple models created for each combination of store-SKU. Going forward, a store-SKU combination would be called a Product Key.

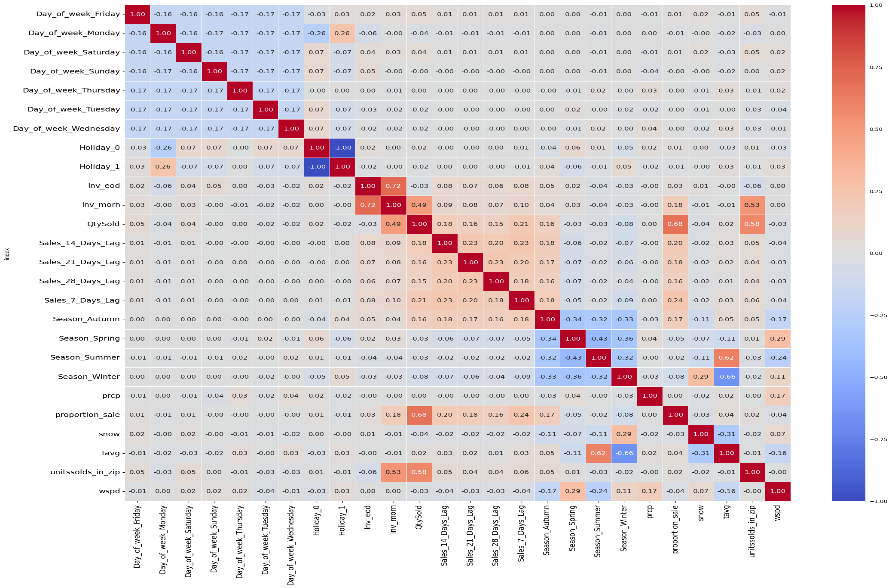
Deep diving into the method/ steps, various modules are used in developing the solution:

1. Preprocessing module:

* Data is captured from a scan-based system. So, a SKU is not necessary to be sold on all the days at a store. Thus, it leads to a point where the data in the system for a SKU does not have it captured on every single day. So, a module is created to impute zeros to no sale days.
* Few SKUs are sold in high quantities on some days and there are also negative sale values recorded on some days which act as outliers to the model, thus an outlier treatment is done to remove these outliers.
* Since the use case is about retail sale at stores, It is a great idea to understand the weather condition in the area to predict the sale. Thus, to collect the weather information, a weather information extraction module is created. Here, Weather information is extracted using a package called metostat. This package fetches the information using metostat API.
* Extracted weather information is merged to the cleaned data to input to the model.

1. Significant variables identification:

* After Preprocessing, data is ready to be trained. However, for better performance, significant factors that impact the prediction need to be identified for better model performance. Thus, Variable importance score provided by random forest algorithm is used to observe the importance score for each variable.
* As there are multiple models created, there would be different significant variables for each Product key. One might have to ponder on how to aggregate the results. To overcome this, a majority vote-based mechanism is chosen to identify the significant variables.
* Significant variables for each model are ranked and the top 5 variables for each model are picked and then the top 5 variables which are significant for most of the models are picked for modelling.
* Along with variable importance scores, correlation matrix and heat maps are derived to visualize and understand the significant factors that influence the prediction of sale. Below is a heat map used in the model development.



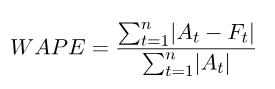
1. Model Training/testing/validation:

* The model uses past one and half year data to predict the sales for the next 3 months. During this step, different hyperparameters of the random forest module are experimented to get acceptable range of model accuracy.
* Of all, tweaking max\_features parameter of random forest yielded best results. Thus max\_features value is set to 0.5 to run the model.
* Again, as the approach is creation of one model for each Product key, value of 0.5 is used for each combination of store-SKU.
* The past 1.5 years data is split to 80% for training and 20% validation.
* Test data of 3 months is passed to model that is validated using respective store-SKU combination data.

1. Model evaluation:

* Here, Evaluation could have been simpler if there was only 1 model that is used to train the data but since there are multiple models – one model for each store-SKU combination, model evaluation is not a straightforward process.
* There are 2 evaluation metrics used to evaluate – WAPE and perc of Percentage of matching days. These are used to classify a developed model as Good/Bad.
* Any model that has a WAPE value less than 40th percentile of the actual sale values is flagged good as a 1st step. Then, the percentage of prediction values that match 50% or more with the actual values is calculated. If the percentage value is more than 80% then the model is flagged as good in the second step.
* A model that is flagged as a good model in both the steps is considered a good model. An illustration is provided below for easy understanding.
* Consider the below values 10 values of sale for 10 days:

A table with numbers and symbols

Description automatically generated 

* WAPE can be calculated using the above formula and the value would be 0.125. It explains that predictions vary from actuals by 12.5%. We define any product key having WAPE < 40th percentile of actual sale as a Good Product Key in the 1st step.
* 40th percentile of Actuals is 1 and WAPE is 0.125. So, in the 1st step, this is a good Product Key
* In the second step, Percentage of 50% or more matching days is calculated. A value is set to be 50% or more matching if Abs((predicted value/actual value) – 1)>=0.5
* Value for 50% or more matching comes to be (7/9) = 77.77%. Thus, it can be flagged as a good Product Key even in 2nd step. Since the model is flagged good in both steps, it can be classified as a good model.
* This process of evaluation is done for all the Product Keys and the results are visualized in Power BI.