CS-535 Project: Time series analysis of e-commerce data

Name: Priyanka Prakash Tanpure. (BU ID : B00821027)

**Introduction**:

* Time Series Data Analysis: Time series analysis is a technique that deals with time series data. Time series data means that data is in a series of particular time periods or intervals. Time Series Analysis accounts for the fact that data points taken over time may have an internal structure (such as autocorrelation, trend or seasonal variation) that should be accounted for.
* Time series data prediction or forecasting: Time series forecasting uses information regarding historical values and associated patterns to predict future activity. Most often, this relates to trend analysis, cyclical fluctuation analysis, and issues of seasonality.

**Solution to the problem - model used (SARIMA)**:

* Why SARIMA?
  + The statsmodels library provides an implementation of the classical decomposition method in a function called seasonal\_decompose(). From the review of the plot we can clearly see we have time series with **trend and seasonality** present in it. So, it make sense to use **seasonal ARIMA** Model. (Fig 1 – page 3)
* Seasonal Autoregressive Integrated Moving Average, SARIMA or Seasonal ARIMA, is an extension of ARIMA that explicitly supports univariate time series data with a seasonal component.
* SARIMA configuration

Trend Elements: (p,d,q)

p: Trend autoregression order ,

d: Trend difference order,

q: Trend moving average order.

Seasonal Elements: (P,D,Q,m)

P: Seasonal autoregressive order,

D: Seasonal difference order,

Q: Seasonal moving average order,

m: The number of time steps for a single seasonal period. (m=7 for daily data, 12 – monthly, 52 - weekly)

* Applied the SARIMA model on both overall data (per day sum of sales of all products) and individual products. If the seasonality present in overall data then the seasonality is present in individual key product data as well.
* **Top-Down Forecast** - Top-down approaches leverage structure that exists in higher-level aggregate data to improve forecasts at lower levels of the forecasting hierarchy.

Top-down sales forecasting begins with combined data on sales of all products. Then it applies the method of statistics to predict sales of individual items.

* Considering Top-Down approach, first found the model for overall data and then applied the same model to the individual key product for predicting future sales.

**How to train model using the given dataset**

* Find Trend(p,d,q) and seasonal(P,D,Q,m) elements : The Akaike information criterion (AIC) is an estimator of the relative quality of statistical models for a given set of data. The AIC value will allow us to compare how well a model fits the data.
* The **pyramid-arima library** for Python allows us to quickly perform this grid search and even creates a model object that you can fit to the training data.

This library contains an **auto\_arima function** that allows us to set a range of p,d,q,P,D,and Q values and then fit models for all the possible combinations. Then the model will keep the combination that reported back the best AIC value.

from pmdarima.arima import auto\_arima

model = auto\_arima(dataset0, start\_p=1, start\_q=1, max\_p=3, max\_q=3, m=7, start\_P=0, start\_Q=0, seasonal=True, d=1, D=1, trace=True, error\_action='ignore', suppress\_warnings=True,

stepwise=True)

**Final evaluation result based on given dataset**

* For Overall data :- Best Model - ARIMA(1,1,1)(0,1,1)[7] : AIC=1512.181
* Applied same model on individual product. (Top-Down Approach)

**Case Study:**

* Relationship between the minimal amount of training data and the prediction accuracy
  + As the amount of training data increases the prediction accuracy increases. The model training and model selection should be done on the fly.
  + Made some experiments with the size training data and calculated the mean absolute percentage error (MAPE) for each set.
  + First trained data size - 80 days , then the Best model was ARIMA(1,1,1)(0,1,1)[7] with MAPE = 34.20 (Fig 2 – page 3)
  + Trained data size – 90 days then the Best model was: ARIMA(1,1,1)(0,1,1)[7] with MAPE : 21.45 (Fig 3 – page 3)
  + Trained data – 110 days then the Best model was: ARIMA(1,1,1)(0,1,1)[7] with MAPE : 9.52 (Fig 4 – page 3)
  + From above observations we can say that mean absolute percentage error decreases with amount of data which means accuracy increases even though we are getting same model values.
  + When the size of train data increases, the model is well trained and the predictions are more closer to actual values of the parameters, thereby increase the accuracy of forecasting.

Fig 1 : Trend and seasonality plot

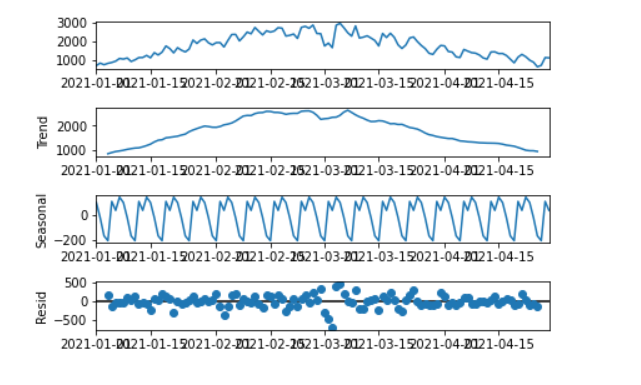


Fig 2 : trained data size - 80 days

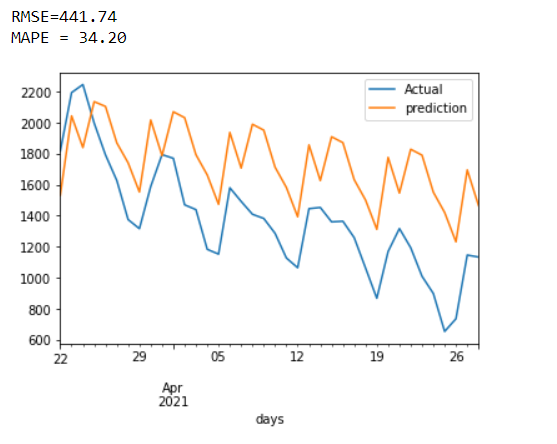


Fig 3 : trained data size - 90 days

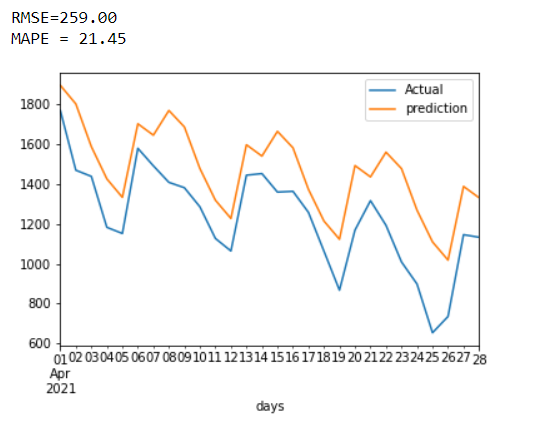
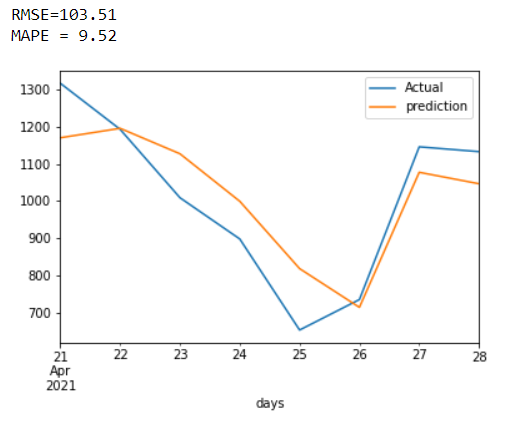
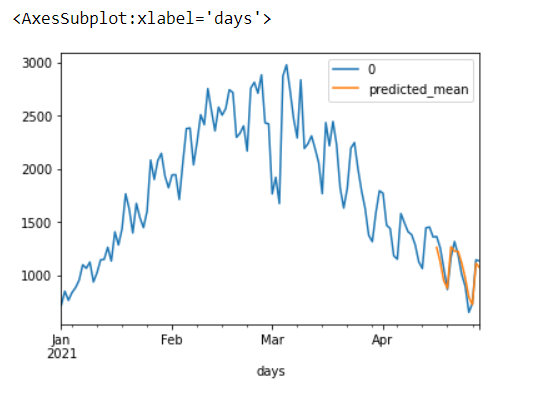


Fig 4: trained data size - 110 days



Overall graph with model ARIMA(1,1,1)(0,1,1)[7] 

References :

* <http://www.forecastpro.com/Trends/forecasting101January2009.html>
* <https://otexts.com/fpp2/top-down.html>
* <https://www.pluralsight.com/guides/advanced-time-series-modeling-(arima)-models-in-python>
* <https://medium.com/@josemarcialportilla/using-python-and-auto-arima-to-forecast-seasonal-time-series-90877adff03c>
* <https://www.analyticsvidhya.com/blog/2015/12/complete-tutorial-time-series-modeling/>
* <https://machinelearningmastery.com/sarima-for-time-series-forecasting-in-python/>
* <https://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/>