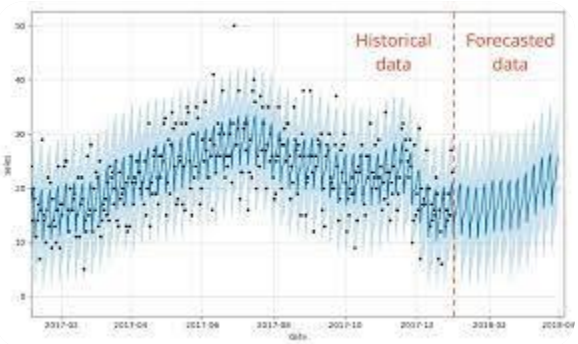

PHASE 1

ABSTRACT

Phase 1: Literature survey & Data Acquisition

Soumen Chatterjee

Phase 1 – 04-Dec-2021

Objectives

Client “Walmart” reached out, to get help on predicting stocks, required to get inventory up to date.

As per the briefing, in 27 countries Walmart operates 11,450 stores, manages inventory across varying climates and cultures.

- Issue-

  Extreme weather events, like hurricanes, blizzards, & floods, have huge impact on sales at the stores and product level. Walmart need a solution which accurately predict the sales of 111 potentially weather-sensitive products (like umbrellas, bread, and milk) around the time of major weather events at 45 of their retail locations which are spread across 20 weather stations. Intuitively, we may expect an uptick in the sales of umbrellas before a big thunderstorm.

- Requirement-

  It's difficult for replenishment managers to correctly predict the level of inventory needed to avoid being out-of-stock or overstock during and after that storm or any other adverse natural events. Walmart has been relying on a variety of vendor tools to predict sales around specific weather events, but it's an ad-hoc and time-consuming process that lacks a systematic measure of effectiveness.

  It's required to create systematic approach which will help Walmart better predict sales of weather-sensitive products.

- Why it’s important –

  Creation of systematic approach using newest technologies backed up scientific & data driven reasoning is very much important for a client like Walmart, this will help them better predict sales of weather-sensitive products. Correct prediction will help replenishment managers to keep the stocks up to date on specific time of a year. If the forecast is too high, it may lead to over-investing and therefore losing money. If the forecast is too low, it may lead to under-investing and therefore losing opportunity.

- Business/Real-world impact of solving this problem –

  As of now it is requested to focus on only 45 retail locations and on 111 products. If the new approach gets successfully into production and correctly predicts then inclusion of other stores, products across the countries will results huge profit to Walmart. As correct prediction will help them plan better in logistics & will help them keep customer base strong.

Dataset:

- Source

  As per the briefing, it's requested to use the data provided by Walmart. A zipped file namely "*Walmart-recruiting-sales-in-stormy-weather*" is received to deal with this specific requirement. The Zipped file contains below mentioned details.

  Data size is well within the scope.

  And being it's structured it can be dealt with standard libraries/tools like Pandas, Matplotlib, Scikit-learn, SQL

a. "weather.csv"

- This file contains details of weather on a specific day/date, starting from 1st Jan 2012 to 31st Oct 2014.
- Consisting of 216435 rows & 20 columns/features.

Feature No.	Feature Name	Description
F1	station_nbr	Weather station number, representing one of 20 weather stations
F2	date	DATE - MM/DD/YYYY
F3	tmax	Maximum Temparature Fahrenheit
F4	tmin	Minimum Temparature Fahrenheit
F5	tavg	Average Temparature Fahrenheit
F6	depart	Departure from normal, that is above/below 30 years' normal
F7	dewpoint	Average dew point
F8	wetbulb	Average wet bulb
F9	heat	Heating (season begins with july)
F10	cool	Cooling (season begins with january)
F11	sunrise	Sunrise (calculated, not observed)
F12	sunset	Sunset (calculated, not observed)
F13	codesum	Weather Phenomena - with 448
F14	snowfall	Height in INCHES
F15	preciptotal	Inches (24-hr period ending at indicated local standard time)
F16	stnpressure	Average Station pressure
F17	sealevel	Average Sea level pressure
F18	resultspeed	Speed in miles per hour
F19	resultdir	Direction to tens of degrees
F20	avgspeed	Average wind speed

b. “train.csv”

- This file contains Date, Store number (an id representing one of the 45 stores), Item number (an id representing one of the 111 products), Unit information (the quantity sold of an item on a given day)
- Consisting of 1048575 rows & 4 columns/features.

c. “key.csv”

- This file contains the mapping of Store number & station number
- Consisting of 1035 rows & 2 columns/features.

d. “sampleSubmission.csv”

- This file contains the input format “**StoreNbr_ProdNbr_Date**” which need to be passed to production unit for prediction
- Consisting of 526917 rows & 2 columns – ID, Units

e. “test.csv”

- Don’t have password to open this file
- This file should have similar structure like b. “train.csv”, & should contains Date, Store number, Item number, unit information
- Consisting of 4 columns/features.

NOTE:

- Data type, description & other details of these files/data will be further explored at the time of Exploratory data analysis stage.

- The Data received should be adequate to do the analysis & prediction

- If required and/or if we don’t have the password to open “test.csv” then we can generate synthetic Data accordingly

Key Performance Indicator

Having a quantifiable activity used to measure how a key aspect of business is operating or how correctly future sales are being predicted in this case, it's important & needed.

Like other usual business cases, here in this scenario as well, we need to think of 4 aspects of KPI and implementing the same, that is implementation of KPI following 4 steps –

Timeframe – Here we will need to analyze the data first and need plan a window of time, like in this case we can consider 3 months window

Measurement/Business Metric – Choosing a business matric, here in this case we can plan for Root Mean Squared Logarithmic Error (RMSLE).

Monitor according to timeframe & improve by the time.

- Business Metric

  As per the requirement stated above it is requested to predict the sales of a specific product in near future using the past data, so it's a Regression problem.

  There are few Metrics which are used to measure the performance of a regression model as mentioned below –

Id	Metric	Aplication
MAE	Mean absolute error	Measure of difference between two continuous variables. MAE is the average
MSE	Mean squared error	Measure the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value
RMSE	Root mean squared error	Frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed. The RMSD represents the square root of the second sample moment of the differences between predicted values and observed values or the quadratic mean of these differences
MAPE	Mean absolute percentage error	Measure of prediction accuracy of a forecasting method in statistics, for example in trend estimation, also used as a loss function for regression problems in machine learning. It usually expresses accuracy as a percentage
RMSLE	Root Mean Squared Logarithmic Error	RMSLE is usually used when you don't want to penalize huge differences in the predicted and the actual values when both predicted and true values are huge numbers.
R-Squared	Coefficient of determination R^2	Is the proportion of the variance in the dependent variable that is predictable from
Adjusted R-Squared	R-Squared with adjust to features quantity	Adding new features to the model, the R-Squared value either increases or remains the same. R-Squared does not penalize for adding features that add no value to the model. So an improved version over the R-Squared is the adjusted R-Squared.

Ref: Image 1- copied from https://towardsdatascience.com/metrics-and-python-850b60710e0c

The problem we are dealing with here - given some data in time, we want to predict the dynamics of that same data in the future. Data, which is shared, contains series of data points indexed (or listed or graphed) in time order. Which is a sequence taken at successive equally spaced points in time. Thus, it is a sequence of discrete-time data.

Hence on its core, this is a time series problem.

And in Timeseries problems we usually use - Root Mean Squared Error (RMSE) and Root Mean Squared Logarithmic Error (RMSLE).

In this case, Root Mean Squared Logarithmic Error (RMSLE) will be used as business metric. As it is better compared to RMSE (details shared below)-

- Mathematical Definitions:

$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2},$	$\text{RMSLE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\log y_i - \log \hat{y}_i)^2}$

Ref: Image 2- copied from Datascience stackexchange

Where y^i is the target value for example i and y^i is the model's prediction. Both the metrics quantify the prediction error, so in general a high RMSE implies a high RMSLE as well.

RMSLE has the meaning of a relative error, while RMSE is an absolute error.

RMSE will have a drastic effect of outliers on its values. But in case of RMSLE we can reduce the effect of outliers by many magnitudes & their effects is much less.

RMSLE value will only consider the relative error between Predicted and the actual value neglecting the scale of data. But RMSE value will increase in magnitude if the scale of error increases. For e.g.

```
Actual value = 100
Predicted Value = 90
RMLSE: 0.1053
RMSE: 10

Actual value = 1000
Predicted Value = 900
RMSLE: 0.1053
RMSE : 100
```

Also, in case of under-estimation results from RMSLE are affected greatly. So, one can easily understand that it is better than RMSE in certain scenarios, but RMSE works better for generalize cases. At last, RMSLE is bit better than RMSE but based upon certain cases only.

Both the metrics that is Root Mean Squared Error (RMSE) and Root Mean Squared Logarithmic Error (RMSLE) can be used in other Timeseries problems where regression models are used like –

Weather forecasting, Stock price predicting, Unemployment for a state each quarter, Average price of gasoline etc.

- Code: Implementation of RMSLE metric using Python (used “math”, “numpy” library)

```
#Importing standard library math
import math
import numpy as np
#creating custom function for RMSLE calculation
def My_RMSLE(pred, trgt):
  #initializing tot variable, which will hold Total value
  tot = 0

  #Looping through the passback variable pred
  for k in range(len(pred)):
    #for each predicted value and corresponding target value is picked below
    #logarithm applied
    LPred= np.log1p(pred[k]+1)
    LTarg = np.log1p(trgt[k] + 1)

    #checking if none of the Pred & Target value is null then
    #squaring the differences
    if not (math.isnan(LPred)) and  not (math.isnan(LTarg)):
       tot = tot + ((LPred-LTarg) **2)

  #finding our average value and then doing Sqrroot & picking final value
  tot = tot / len(pred)
  return np.sqrt(tot)

y_pred = [2,5,6,1,7,9]
y      = [2.5,6,5,1,7,8.5]
print ('My custom RMSLE: ' + str(My_RMSLE(y_pred,y)))
```

Real world challenges and constraints

- Challenges, constraints foreseeing & the requirement

  Being this problem is a Timeseries & forecasting related we will need to check whether the data is stationary, seasonality, autocorrelated or not.
  Using Dickey-Fuller test, which is a statistical test that we will need to run to determine if the data received is stationary or not.
  If it's not stationary, then we will need to transform to make them stationary.
  As ideally, we want to have a stationary time series for modelling.

  And after that we will need to pick a right Model like- Moving Average, Exponential Smoothing, ARIMA, SARIMA to meet the final requirement to predict the further sales based on available past data.

Similar problems & references which can be referred

- It's a Regression problem as we need to predict sales (continuous value) of a product in near future using available past data

- The problem we are dealing with here - given some data in time, we want to predict the dynamics of that same data in the future. Data, which is shared, contains series of data points indexed (or listed or graphed) in time order. Which is a sequence taken at successive equally spaced points in time. Thus, it is a sequence of discrete-time data.
  Hence on its core, this is a time series problem.

- We can also refer other similar kind of problems like - Closing price of a stock each day, Product sales in units sold each day for a store, Unemployment for a state each quarter, The average price of gasoline each day, Forecasting Weather, Traffic, Churn, Text Generation etc.

- Other than using standard models like – AR, MA, ARMA, ARIMA, SARIMA, SARIMAX, VAR, VARMA, VARMAX, SES, HWES, we can try using Deep learning for predicting.

- Recurrent Neural Networks are the most popular Deep Learning technique for Time Series Forecasting since they allow to make reliable predictions on time series in many different problems. The main problem with RNNs is that they suffer from the vanishing gradient problem when applied to long sequences.

References

1. https://towardsdatascience.com/metrics-and-python-850b60710e0c
2. https://www.quora.com/What-is-the-difference-between-an-RMSE-and-RMSLE-logarithmic-error-and-does-a-high-RMSE-imply-low-RMSLE
3. https://onstrategyhq.com/resources/kpis-vs-metrics-tips-tricks-to-performance-measures/
4. https://datascience.stackexchange.com/questions/63514/what-is-the-difference-between-an-rmse-and-rmsle-logarithmic-error
5. https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775
6. https://towardsdatascience.com/sales-forecasting-from-time-series-to-deep-learning-5d115514bfac
7. https://www.tableau.com/learn/articles/time-series-forecasting
8. https://machinelearningmastery.com/time-series-forecasting-methods-in-python-cheat-sheet/
9. https://towardsdatascience.com/time-series-forecasting-with-deep-learning-and-attention-mechanism-2d001fc871fc