OPTIMIZATION OF A BUSINESS MANAGEMENT SYSTEM

ABUAYYASH Hosni, IBADOF Alin-Gabriel
Faculty: Industrial Engineering and Robotics, Specialization: Industrial Engineering,
Year: 2nd, E-mail: Hosni.ayyash@gmail.com

ABSTRACT: The aim of this paper is to embed knowledges of Modelling & Simulation, Economics and Databases into one application regarding business management system optimization to apply what we have learned and see the integrity of them together as a whole. This paper takes the idea of a business model that should help a business production line to operate efficiently and deliver better quality services to their customers by pointing a few concepts we have learned in the mentioned disciplines. The power of databases on the Cloud and the services of IBM's "Watson Studio" was used to present these papers. In order to reach our objective, we have simulated an interactive unit for a business production line where we needed to populate with data for a whole year so we can apply some analytical statistics in order to help the management of the business to take better decisions. [6], [7]

KEYWORDS: economics, business management, databases, optimization, PED

1. Introduction

The focus of this paper is a production line database which has been simulated to integrate the orders, stocks of raw materials and logistics, labour, suppliers, and sales to build a virtual model. By creating an algorithm to generate orders for one year for both production lines in this virtual model, we took in concern the integrity functions of the whole system by using SQL language. The purpose of this is to subtract reports of data for the whole year (employees, sales, costs, revenue, profits, taxes, and prices). The next step was to model a few views out of this data in order to show the management of this production line some economic aspects like salary checks based on the current Romanian system, the diverse sales taxes depending on the raw materials, highlighting the variable cost and fixed costs. And at last, Modelling & Simulation ideas were used to simulate sales for 23 years on Microsoft's Excel [8] and to connect it back to the relevant database where Python has been used to see the effect of changing the price on the quantity sold in purpose to increase the revenues by applying the concepts of economics "Price Elasticity of Demand" and "Revenue elasticity of Demand" through tables and graphs. Although the we have created on our own the database for the simulation, as well as the calculations, we needed to use an open-source code for Python to plot the graph (Price vs. Quantity) [4].

2. Modules

After carefully analysing a few small to middle-size business management systems we have gathered the required information for building our database which contains 16 main tables and another 8 auxiliary tables to help create the algorithm for populating the modelled database.

The stocks tables have been created to handle the raw materials needed in the production line as well as the logistics and housekeeping.

Connected to these stocks, are two lists of products offered for sale through an order table where the employees are placing the order by using a trigger that provides the ability to check the availability of raw materials in the stock and then subtracting these quantities from stocks and placing the equivalent quantities in a demanding table in advance for next week's order to the suppliers with the purpose of refilling the stock again to keep the flow of production. By the end of this operation, the mentioned trigger will assure the registering of the ordered product into the receipt table on the spot with the specific date and time.

The employee table has been created to hold information about name, salary, wage, position, address, IBAN, and the contact data. Again, this table is a part of a system that is connected with the objective of assuring the flow of production (placing orders, performing the production line's operation, receiving from supplier and of course managing the production system) while respecting a schedule determined by a check-in – check-out table (this table also helps to calculate the salaries at the end of each month).

Algorithms used for generating orders took in concern the spatial and human resources capacity on the location and the time related to the production and services provided day by day in a rational way where it can be that certain days are more busy than others, as well as a period of each day, and at last the season of the year. These algorithms use a different trigger to operate the orders because they need to randomize the mentioned times, which prevent using the regular trigger.

Implementation of this production line model was easier by using the knowledge gained throughout the Modelling & Simulation course where we have learned there is no perfect system, and always there will be a margin of error which we took in consideration while building our algorithms for populating the production database. The course also offered us the tools for modelling the data which is required to perform some analytical reports and applying, of course, few concepts we have learned in economics. We used this because our populated database performed a production operation just for one year while we needed data for many years in order to test the effect of changing the prices of the business unit on the quantity sold with the purpose of making the right decision and as a result making more revenue.

The output reports of the populated data in the production database have been established respecting the economic aspects (See Fig. 1)

- Price
- Cost
  - Fixed cost
    - Rent
    - Utilities
    - Salaries
    - Housekeeping
  - Variable cost
    - Raw materials
    - Tax of production line 1, 2
    - Error of production line 1, 2
- Revenue
  - Revenue production line 1, 2
- Profit

DATE	REVENUE_KITCHEN	COST_KITCHEN	TAX_KITCHEN	ERROR_KITCHEN	REVENUE_BAR	COST_BAR	TAX_BAR	ERROR_BAR	FIX_COST	PROFITE
2019-03-04	3017.00	956.9990	150.8500	120.52	2095.00	515.1400	361.2300	71.74	2099.654400	1822.166600
2019-03-05	3052.00	1069.4930	152.6000	153.98	2037.00	518.1600	348.2500	32.86	2099.654400	1701.702600
...										
2020-02-29	5258.00	1661.2680	262.9000	140.92	3881.00	924.4200	661.0900	84.10	2215.344000	4203.458000

Fig. 1. Sales Report

Revenue calculated:

$$Revenue1 = quantity\ sold1 * price \quad (1)$$
$$Revenue2 = quantity\ sold2 * price \quad (2)$$

Tax:

$$Tax1 = revenue * 0.05 \quad (3)$$
$$Tax2 = revenue * 0.19 \quad (4)$$

Profit:

$$Profit = Revenue1 + Revenue2 - Variable\ cost - Fixed\ cost - Tax1 - Tax2 \quad (5)$$

Fixed cost:

$$Fixed\ cost = \frac{((utilities + rent + salaries + housekeeping\ cost)\ per\ month)}{no.of\ the\ month's\ working\ days} \quad (6)$$

Cost is calculated by summing the cost of raw materials per unit produced. Price is established based on the list of products. Error cost is calculated by summing the cost of raw materials per unsold unit. Salaries have been calculated as the way done in the Romanian salary ticket (See Fig. 2).

	DATE	Name	Wage/Hour	No_hours	SALARY_BRUT	CASS	CAS	RETINUT	BAZA	IMPOZIT	REVINE
0	2019-03	Hosni Abuayyash	45.45	160	7272	727	1818	2545	4727	473	4254
1	2019-03	Alin Ibadof	32.95	176	5799	580	1450	2030	3769	377	3393

.....

14	2019-03	Anne Marie	15.91	168	2673	267	668	936	1737	174	1564
15	2019-03	Lionel Reeves	15.91	168	2673	267	668	936	1737	174	1564

Fig. 2. Salary tickets table

$$Wage/Hour = \frac{salary\ per\ month}{176\ hours} \quad (7)$$
$$No\_hours = \sum daily\ checked\ in\ and\ checked\ out\ hours \quad (8)$$
$$Salary\_brut = wage/hour * no\_hours \quad (9)$$
$$CASS = salary\ brut * 0.10 \quad (10)$$
$$CAS = salary\ brut * 0.25 \quad (11)$$
$$Retinut = CASS + CAS \quad (12)$$
$$BAZA = salary\ brut - retinut \quad (13)$$
$$Impozit = baza * 0.10 \quad (14)$$
$$Revine = baza - impozit \quad (15)$$

Statistical description for the whole year sales report, shown in Fig. 3.

	REVENUE_KITCHEN	COST_KITCHEN	TAX_KITCHEN	ERROR_KITCHEN	REVENUE_BAR	COST_BAR	TAX_BAR	ERROR_BAR	FIX_COST	PROFITE
count	363.000000	363.000000	363.000000	363.000000	363.000000	363.000000	363.000000	363.000000	363.000000	363.000000
mean	3430.024793	1196.901116	171.501240	101.097576	2409.129477	580.623278	415.437631	56.368044	2230.845752	2085.285970
std	1032.893988	364.875792	51.644699	44.917280	734.397658	176.615029	128.332714	21.416697	71.777349	1047.968691
min	1305.000000	432.279000	65.250000	9.000000	1034.000000	238.590000	175.230000	16.400000	2106.654400	79.807000
25%	2915.500000	991.968500	145.775000	70.060000	2017.000000	476.240000	340.030000	40.745000	2210.344000	1662.206000
50%	3346.000000	1166.021000	167.300000	95.960000	2356.000000	570.250000	406.750000	55.350000	2226.344000	2016.666000
75%	3807.500000	1359.615000	190.375000	125.020000	2665.000000	645.650000	464.305000	69.755000	2324.033600	2373.504000
max	6472.000000	2393.219000	323.600000	289.060000	4943.000000	1185.230000	858.390000	156.360000	2337.033600	4978.933000

Fig. 3. Statistical report for sales

3. Price Elasticity of Demand (PED)

Price elasticity of demand is an economic measure of the change in the quantity demanded or purchased of a product in relation to its price change. [1]

We used the output data of our simulated business production line to estimate Price Elasticity of Demand and the Price Elasticity of Revenue. Therefore, because we have populated this simulation for one-year orders, it was required to model more data for another 22 years, where we have done the following:

- ⇨ Price: We have 18 items on the product list offered for sale from which we have taken the mean of the maximum and the minimum price;
- ⇨ Quantity sold: the output of the generated data was for one year, depending on counting the timestamp from each receipt. (See Fig. 4)

YYYY-MM	Sum of Quantity Sold
2019-03	5900
2019-04	6317
2019-05	6442
2019-06	3868
2019-07	3902
2019-08	3997
2019-09	6318
2019-10	6468
2019-11	6389
2019-12	6462
2020-01	6433
2020-02	6227
Grand Total	**68723**

Fig. 4. Quantity Sold per month

The mentioned data has been moved through a Comma Separated Values file (.csv), and then by using a pivot table on Excel we got a summarization for the quantity sold each quarter of the year. Using this we have calculated the mean of those quarters and we plugged it to the top of a modelled table as the first quarter for the year 1997 followed with the mean price.

We decided the value 1.0124 for price percentage incrementation and we added to it the "Random" Excel built-in function multiplied with 0.5, multiplied again to a (-1,1) "Randombetween" function.

For quantity sold, the value 0.988 has been decided as a decremented percentage multiplied with the quantity from the modelled table in addition to "Randombetween(-1000,1000)" function. The output value was rounded to an accuracy of 0 decimals.

Both equations have been extended starting from the first season of 1997 until the last season of 2019, as well as containing the corresponding year and season number.

This table has been uploaded to the IBM's Cloud, where our simulated production database is located.

At the end, the connection to IBM's Watson Studio service was established where we used a Python notebook to perform some functions to display the concept of price elasticity of demand, through tables and graphs, the outputs shown in table of Fig. 5 and graph of Fig. 6.

	year	Quarter_x	QUANTITY	price
0	1997	1	17243	49.0000
1	1997	2	16425	49.1217
2	1997	3	15736	49.6059
3	1997	4	15526	50.5604
...	...	...	...	...
91	2019	4	5710	145.3714

Fig. 5. The generated model of the changing in price and quantity seasonally through 23 years

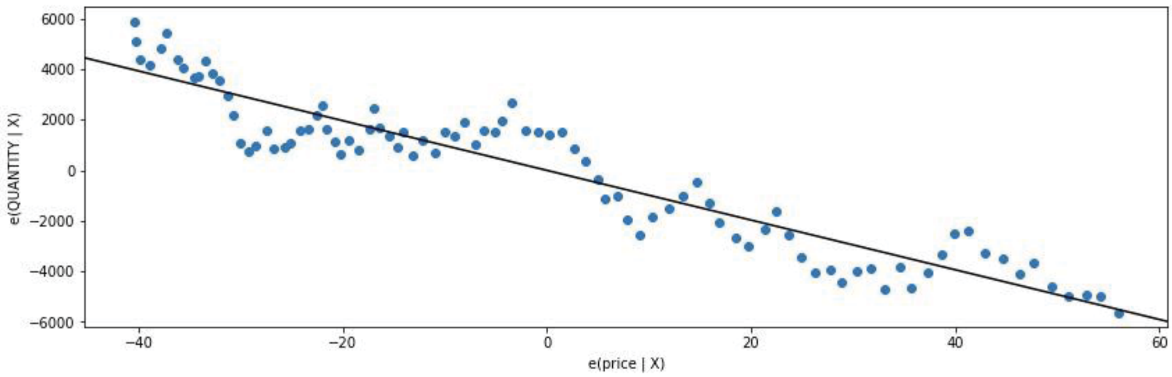

Fig. 6. Graphical representation of price corresponding to quantity

To demonstrate the concept of price elasticity of demand (PED) is required to calculate the price through equation (17)

$$PED = \frac{\%Q_{sold}}{\%P} = \frac{\frac{\Delta Q_{sold}}{Q_{soldo}}}{\frac{\Delta P}{P_0}} = \frac{\Delta Q_{sold}}{\Delta P} \cdot \frac{P_0}{Q_{soldo}} \quad (16)$$

After that, revenue has been calculated for each season using the following equation:

Revenue = price * quantity sold (17)

The resulting revenue values have been added as a column to the table mentioned before. To find out the price elasticity of revenue (PER), we have used the following formula:

$$PER = \frac{\%R}{\%P} = \frac{\frac{\Delta R}{R_0}}{\frac{\Delta P}{P_0}} = \frac{\Delta R}{\Delta P} \cdot \frac{P_0}{R_0} \quad (18)$$

Where "R" refers to the revenue and "P" refers to the price. The result is the following:

Date	QUANTITY	price	PED	Revenue	RED
1997-03-03	17243	49.0000	NaN	8.449070e+05	NaN
1997-06-02	16425	49.1217	-19.100555	8.068239e+05	-18.147994
1997-09-01	15736	49.6059	-4.255616	7.805984e+05	-3.297564
1997-12-01	15526	50.5604	-0.693557	7.850008e+05	0.293098
...	...	...	...	...	...
2019-12-02	5710	145.3714	-8.525656	8.300707e+05	-7.631373

Fig. 7. Price Elasticity of Demand and Revenue Elasticity of Demand along 23 years

5. Conclusion

Price elasticity of revenue is an important parameter like the price elasticity of demand. No matter what a production line business is, when there is a discussion about the revenue, any management in charge's desire is to choose the price that maximizes the revenue and consequently the profit. For this purpose, we have decided to describe the price elasticity of demand which is defined as the percentage change in price divided on the percentage change in quantity sold.

6. References:

[1]. WILL KENTON (2020), Price Elasticity of Demand, https://www.investopedia.com/terms/p/priceelasticity.asp, Accessed on 07.05.2020

[2]. https://en.wikipedia.org/wiki/Elasticity_(economics), Accessed on 07.05.2020

[3] https://en.wikipedia.org/wiki/Price_elasticity_of_demand, Accessed on 07.05.2020

[4]. Susan Li (Sep 1, 2018), Price Elasticity of Demand, Statistical Modeling with Python, https://towardsdatascience.com/calculating-price-elasticity-of-demand-statistical-modeling-with-python-6adb2fa7824d, Accessed on 07.05.2020

[5]. https://en.wikipedia.org/wiki/IBM_cloud_computing, Accessed on 07.05.2020

[6]. https://www.ibm.com/cloud/db2-on-cloud/, Accessed on 07.05.2020

[7]. https://www.ibm.com/cloud/watson-studio, Accessed on 07.05.2020

[8]. https://en.wikipedia.org/wiki/Microsoft_Excel, Accessed on 07.05.2020

7. Abbreviations

The following symbols are used in the work:

PED = Price Elasticity of Demand

PER = Price Elasticity of Revenue

CASS = Contribuția de asigurări sociale

CAS = Contribuția de asigurări sociale de sănătate

Q_{sold} = Quantity Sold

Q_{sold0} = Initial Quantity Sold