OPTIMIZATION OF THE AVERAGE MONTHLY COST OF AN EOQ INVENTORY MODEL FOR DETERIORATING ITEMS IN MACHINE LEARNING USING PYTHON

by

KALAIARASI, K. [a,b]***, SOUNDARIA, R.*** [a]***, Nasreen KAUSAR*** [c]***, Praveen AGARWAL*** [d]***, Hassen AYDI*** [e,f,g]***, and Habes ALSAMIR*** [h*]

[a] PG and Research Department of Mathematics, Cauvery College For Women (Autonomous), Affilated to Bharathidasan University, Tiruchirappalli, Tamil Nadu, India
[b] Department of Mathematics, Srinivas University, Suranthkal, Mangalore, Karnataka, India
[c] Department of Mathematics, Faculty of Arts and Science, Yildiz Technical University, Esenler, Istanbul, Turkey
[d] Department of Mathematics, Anand International College of Engineering, Jaipur, Rajasthan, India
[e] Institut Superieur d'Informatique et des Techniques de Communication, Universite de Sousse, H. Sousse, Tunisia
[f] Department of Mathematics and Applied Mathematics, Sefako Makgatho Health Sciences University, Ga-Rankuwa, South Africa
[g] China Medical University Hospital, China Medical University, Taichung, Taiwan
[h] College of Business Administration-Finance Department, Dar Al Uloom University, Al Falah, Riyadh, Saudi Arabia

Original scientific paper
https://doi.org/10.2298/TSCI21S2347K

In many stock disintegration issues of the real world, the decay pace of certain things might be influenced by other contiguous things. Depending on the situation, the influence of weakened items can be reduced by eliminating them through examination. We specify a model that impacts the average monthly cost, and the non-linear programming Lagrangian method is solved the specified model. The fuzzify inventory model is used to determine the lowest cost by employing a trapezoidal fuzzy number, and the defuzzification process is performed using the graded mean integration representation method. To test the model, we created a CSV file, used PYTHON (version 3.8.5), we developed a program to predict the economic order quantity and total cost.

Key words: *economic order quantity, optimal total cost, machine learning, trapezoidal fuzzy number, Lagrangian method, PYTHON, graded mean integration representation method*

Introduction

Inventory is related to decisions that minimize the average total cost or maximize the solution of the average total profit. In inventory models, deterioration plays a significant role. Deteriorate is defined as damage to inventory quality. Food, medicine, vegetables, *etc.* are classified as *deteriorating*. In the inventory process, the project has been lost many times during the deterioration process, so this loss must be considered when analyzing the system. Ghare and Schrader [1] introduced the concept of optimal ordering policies for deteriorating items. Covert

* Corresponding author, e-mail: habes@dau.edu.sa

and Philip [2] developed a deterioration model for Weibull distribution. Dave and Patel [3] developed a deterioration model for linear demand, Misra [4] formulated variable rate of deterioration, and Sachan [5] extended it with shortages. Dye *et al.* [6] identified an appropriate selling price and lot size when partial backlog is accompanied by varied rates of deterioration. The first to introduce inspection was Salameh and Jaber [7], who utilized an inventory model that added a defect rate to each suborder. The economic order quantity (EOQ) is that combines time-dependent demand under inflation with deteriorating items. Ben-Daya *et al.* [8], introduced an incorporated Stock model that requests Amount and investigation types considered choice factors. There was no evaluation, a 100% inquiry, and a testing examination to resolve it. In the condition of testing examination, the investigation was exposed to both purchaser and provider dangers to accomplish reasonable outcomes. The EOQ is the requested quantity, which limits the absolute storage cost and application cost. Today, the EOQ model is so notable that we acknowledge its essential design as self-evident. Harris [9] invented EOQ in 1913, Wilson, an expert who has widely utilised it, and Andler, on the other hand, are recognized for their work.

Zadeh [10] proposed fuzzy sets as an extension of classic sets. In solving the fuzzy linear programming problem, we define a crisp model to find an optimal solution. Zimmermann [11] introduced the solution of fuzzy linear programming. Kaufmann *et al.* [12], introduced the fuzzy optimal solution of fuzzy non-linear programming Problems with inequality constraints. Chen and Hsieh [13] proposed the concept of graded mean integration representation method (GMIRM). Kalaiarasi *et al.* [14], initiated the concept of PYTHON in the fuzzy inventory model.

Siva Jyothi and Rohit [15] discussed the importance of PYTHON for data science. A PYTHON is an object-familiarized, written, and inferred language for learning. The PYTHON could be a powerful problem-oriented language created by Guido Van Rossum. It permits coders to put in writing code in fewer lines that are unattainable with different languages.

Preliminaries

Definition 1 (The fuzzy arithmetical operations under function principle) [16]

Chen [16] introduced the function principle. The following are some TFN fuzzy arithmetical procedures stated using the function principle: Suppose $\tilde{C} = (c_1, c_2, c_3, c_4)$ and $\tilde{D} = (d_1, d_2, d_3, d_4)$ are TFN. Then:

– $\tilde{C} \oplus \tilde{D} = (c_1 + d_1, c_2 + d_2, c_3 + d_3, c_4 + d_4)$
– $\tilde{C} \otimes \tilde{D} = (c_1 d_1, c_2 d_2, c_3 d_3, c_4 d_4)$
– $\tilde{C} \ominus \tilde{D} = (c_1 - d_4, c_2 - d_3, c_3 - d_2, c_4 - d_1)$
– $\tilde{C} \oslash \tilde{D} = \left(\frac{c_1}{d_4}, \frac{c_2}{d_3}, \frac{c_3}{d_2}, \frac{c_4}{d_1} \right)$
– Let $Z \in \mathbb{R}$, then
  - $Z \geq 0, \quad Z \otimes \tilde{C} = (ZC_1, ZC_2, ZC_3, ZC_4)$
  - $Z \leq 0, \quad Z \otimes \tilde{C} = (ZC_4, ZC_3, ZC_2, ZC_1)$

Definition 2 Graded mean integration representation method (GMIRM) [17]

For defuzzification, the method is based on the integral value of the graded mean h-level of a generalised fuzzy number. The GMIRM of $\tilde{A}$ is defined:

$$P(\tilde{A}) = \frac{\int_0^w h\left(\frac{L^{-1}(h)+R^{-1}(h)}{2}\right)\mathrm{d}h}{\int_0^w h\mathrm{d}h}$$

The GMIRM of $\tilde{B}$ is:

$$P(\tilde{B}) = \frac{\int_0^1 h\left(\frac{b_1+b_4+(b_2-b_1-b_4+b_3)h}{2}\right)\mathrm{d}h}{\int_0^1 h\mathrm{d}h} = \frac{b_1+2b_2+2b_3+b_4}{6}$$

Fuzzy inventory model

Notation applied

Symbols	Description	Unit
O	Fixed ordering cost	Rupees per unit
β	Coefficient of the effect of deteriorated items on desirable items	Rupees per unit
θ_0	Fixed deterioration rate of single items	unit per month
T	Cycle length	month
S	Setup cost	Rupees per unit
C	Purchasing cost	Rupees per unit
D	Annual demand rate	Rupees per unit
h	Holding cost per item	Rupees per month
n	Number of inspections at each period	Rupees per month

Formulation of the inventory model

Consider an organic product distributor, the organization ought to decide its approaches so that the average total cost is minimized. Abolfazl and Gholamian [18] given an inventory model for deteriorating items. The integrated inventory model is given by:

$$JC = \frac{2O}{T} + 0.5TCD + 2\frac{Sn}{T} + 0.5hDT + 0.5T\left(\theta_0 + \frac{\beta}{n}\right)$$

Differentiating partially equation with respect to T, and:

$$\text{Put } \frac{\partial JC}{\partial T} = 0, \quad \text{we get} \quad T = \sqrt{\frac{2(O+Sn)}{0.5CD + 0.5hD + 0.5\left(\theta_0 + \frac{\beta}{n}\right)}}$$

Fuzzy inventory model for JC(T)

Suppose, $\tilde{D} = (D_1, D_2, D_3, D_4)$, $\tilde{S} = (S_1, S_2, S_3, S_4)$, $\tilde{T} = (T_1, T_2, T_3, T_4)$ and by applying Graded Mean Representation we get:

$$JC(T) = \frac{1}{6}\left\{\begin{array}{l} \frac{2O}{T_4} + 0.5T_1CD_1 + 2\frac{S_1 n}{T_4} + 0.5hD_1T_1 + 0.5T_1\left(\theta_0 + \frac{\beta}{n}\right) + \\ +2\left[\frac{2O}{T_3} + 0.5T_2CD_2 + 2\frac{S_2 n}{T_3} + 0.5hD_2T_2 + 0.5T_2\left(\theta_0 + \frac{\beta}{n}\right)\right] + \\ +2\left[\frac{2O}{T_2} + 0.5T_3CD_3 + 2\frac{S_3 n}{T_2} + 0.5hD_3T_3 + 0.5T_3\left(\theta_0 + \frac{\beta}{n}\right)\right] + \\ +\frac{2O}{T_1} + 0.5T_4CD_4 + 2\frac{S_4 n}{T_1} + 0.5hD_4T_4 + 0.5T_4\left(\theta_0 + \frac{\beta}{n}\right) \end{array}\right\}$$

Now differentiate partially with respect to T_1, T_2, T_3, T_4 and equate it to 0 we get:

$$T^* = \sqrt{\frac{12O + 2n(S_1 + 2S_2 + 2S_3 + S_4)}{C(0.5D_1 + D_2 + D_3 + 0.5D_4) + h(0.5D_1 + D_2 + D_3 + 0.5D_4) + 3\theta_0 + 3\frac{\beta}{n}}}$$

Lagrangian method for solving EOQ model

Hamdy [19] discussed how to use the Lagrangian conditions to find the best solution to a non-linear programming problem with inequality constraints. Alsaraireh *et al.* [20, 21], solved the non-linear fuzzy inventory model using Lagrangian method.

Suppose fuzzy order quantity value $\tilde{T}$ be $\tilde{T} = (T_1, T_2, T_3, T_4)$ with $0 < T_1 \le T_2 \le T_3 \le T_4$ from (1) we have $0 < T_1 \le T_2 \le T_3 \le T_4$. We replace inequality conditions $0 < T_1 \le T_2 \le T_3 \le T_4$ into the following inequality $T_2 - T_1 \ge 0$, $T_3 - T_2 \ge 0$, $T_4 - T_3 \ge 0$, $T_1 > 0$. We use the Lagrange method to find minimize $JC(T)$.

Step 1. To find the $\min P[JC(T)]$, Put:

$$\frac{\partial JC}{\partial T_1} = 0, T_1 = \sqrt{\frac{2O + 2S_4 n}{0.5CD_1 + 0.5hD_1 + 0.5\left(\theta_0 + \frac{\beta}{n}\right)}}$$

$$\frac{\partial JC}{\partial T_2} = 0, T_2 = \sqrt{\frac{4O + 4S_3 n}{CD_2 + hD_2 + 0.5\left(\theta_0 + \frac{\beta}{n}\right)}}$$

$$\frac{\partial JC}{\partial T_3} = 0, T_3 = \sqrt{\frac{4O + 4S_2 n}{CD_3 + hD_3 + 0.5\left(\theta_0 + \frac{\beta}{n}\right)}}$$

$$\frac{\partial JC}{\partial T_4} = 0, T_4 = \sqrt{\frac{2O + 2S_1 n}{0.5CD_4 + 0.5hD_4 + 0.5\left(\theta_0 + \frac{\beta}{n}\right)}}$$

From the previous, $T_1 > T_2 > T_3 > T_4$. It is not satisfying the constraint $0 < T_1 \le T_2 \le T_3 \le T_4$.

Step 2. Convert the constraint $T_2 - T_1 \ge 0$ into $T_2 - T_1 = 0$ and the Lagrangian function as $L(T_1, T_2, T_3, T_4, \lambda) = P[JC(T)] - \lambda(T_2 - T_1)$:

$$\frac{\partial L}{\partial T_1} = \frac{1}{6}\left[-\frac{2O}{T_1^2} + 0.5CD_1 + 0.5hD_1 - \frac{2S_4 n}{T_1^2} + 0.5\left(\theta_0 + \frac{\beta}{n}\right)\right] + \lambda = 0$$

$$\frac{\partial L}{\partial T_2} = \frac{2}{6}\left[-\frac{2O}{T_2^2} + 0.5CD_2 + 0.5hD_2 - \frac{2S_3 n}{T_2^2} + 0.5\left(\theta_0 + \frac{\beta}{n}\right)\right] - \lambda = 0$$

$$\frac{\partial L}{\partial T_3} = \frac{2}{6}\left[-\frac{2O}{T_3^2} + 0.5CD_3 + 0.5hD_3 - \frac{2S_2 n}{T_3^2} + 0.5\left(\theta_0 + \frac{\beta}{n}\right)\right] = 0$$

$$\frac{\partial L}{\partial T_4} = \frac{2}{6}\left[-\frac{2O}{T_4^2} + 0.5CD_4 + 0.5hD_4 - \frac{2S_1 n}{T_4^2} + 0.5\left(\theta_0 + \frac{\beta}{n}\right)\right] = 0$$

$$\frac{\partial L}{\partial \lambda} = -(T_2 - T_1) = 0$$

$$T_1 = T_2 = \sqrt{\frac{6O + 2S_4 n + 4S_3 n}{C(0.5D_1 + D_2) + h(0.5D_1 + D_2) + 1.5\left(\theta_0 + \frac{\beta}{n}\right)}}$$

$$T_3 = \sqrt{\frac{4O + 4S_2 n}{CD_3 + hD_3 + \left(\theta_0 + \frac{\beta}{n}\right)}}$$

$$T_4 = \sqrt{\frac{2O + 2S_1 n}{0.5CD_4 + 0.5hD_4 + 0.5\left(\theta_0 + \frac{\beta}{n}\right)}}$$

From the previous, $T_3 > T_4$ it does not satisfy the constraint $0 < T_1 \le T_2 \le T_3 \le T_4$.

Step 3. Convert the constraints $T_2 - T_1 \ge 0, T_3 - T_2 \ge 0$ into $T_2 - T_1 = 0$ and $T_3 - T_2 = 0$. We optimize $P[JC(T)]$ then the Lagrangian method is $L(T_1, T_2, T_3, T_4, \lambda_1, \lambda_2) = P[JC(T)] - \lambda_1(T_2 - T_1) - \lambda_2(T_3 - T_2)$:

$$\frac{\partial L}{\partial T_1} = \frac{1}{6}\left[-\frac{2O}{T_1^2} + 0.5CD_1 + 0.5hD_1 - \frac{2S_4 n}{T_1^2} + 0.5\left(\theta_0 + \frac{\beta}{n}\right)\right] + \lambda_1 = 0$$

$$\frac{\partial L}{\partial T_2} = \frac{2}{6}\left[-\frac{2O}{T_2^2} + 0.5CD_2 + 0.5hD_2 - \frac{2S_3 n}{T_2^2} + 0.5\left(\theta_0 + \frac{\beta}{n}\right)\right] - \lambda_1 + \lambda_2 = 0$$

$$\frac{\partial L}{\partial T_3} = \frac{2}{6}\left[-\frac{2O}{T_3^2} + 0.5CD_3 + 0.5hD_3 - \frac{2S_2 n}{T_3^2} + 0.5\left(\theta_0 + \frac{\beta}{n}\right)\right] - \lambda_2 = 0$$

$$\frac{\partial L}{\partial T_4} = \frac{1}{6}\left[-\frac{2O}{T_4^2} + 0.5CD_4 + 0.5hD_4 - \frac{2S_1 n}{T_4^2} + 0.5\left(\theta_0 + \frac{\beta}{n}\right)\right] = 0$$

$$\frac{\partial L}{\partial \lambda_1} = -(T_2 - T_1), \quad \frac{\partial L}{\partial \lambda_2} = -(T_3 - T_2)$$

$$T_1 = T_2 = T_3 = \sqrt{\frac{10O + 2S_4 n + 4S_3 n + 4S_2 n}{C(0.5D_1 + D_2 + D_3) + h(0.5D_1 + D_2 + D_3) + 2.5\left(\theta_0 + \frac{\beta}{n}\right)}}$$

$$T_4 = \sqrt{\frac{2O + 2S_1 n}{0.5CD_4 + 0.5hD_4 + 0.5\left(\theta_0 + \frac{\beta}{n}\right)}}$$

From the previous, $T_4 > T_1$ it does not satisfy the constraint $0 < T_1 \le T_2 \le T_3 \le T_4$.

Step 4. Convert the constraints $T_2 - T_1 \ge 0, T_3 - T_2 \ge 0$ and $T_4 - T_3 \ge 0$ into $T_2 - T_1 = 0$, $T_3 - T_2 = 0$ and $T_4 - T_3 = 0$. The Lagrangian function is given by $L(T_1, T_2, T_3, T_4, \lambda_1, \lambda_2, \lambda_3) = P[JC(T)] - \lambda_1(T_2 - T_1) - \lambda_2(T_3 - T_2) - \lambda_3(T_4 - T_3)$:

$$\frac{\partial L}{\partial T_1} = \frac{1}{6}\left[-\frac{2O}{T_1^2} + 0.5CD_1 + 0.5hD_1 - \frac{2S_4 n}{T_1^2} + 0.5\left(\theta_0 + \frac{\beta}{n}\right)\right] + \lambda_1 = 0$$

$$\frac{\partial L}{\partial T_2} = \frac{2}{6}\left[-\frac{2O}{T_2^2} + 0.5CD_2 + 0.5hD_2 - \frac{2S_3 n}{T_2^2} + 0.5\left(\theta_0 + \frac{\beta}{n}\right)\right] - \lambda_1 + \lambda_2 = 0$$

$$\frac{\partial L}{\partial T_3} = \frac{2}{6}\left[-\frac{2O}{T_3^2} + 0.5CD_3 + 0.5hD_3 - \frac{2S_2 n}{T_3^2} + 0.5\left(\theta_0 + \frac{\beta}{n}\right)\right] - \lambda_2 = 0$$

$$\frac{\partial L}{\partial T_4} = \frac{1}{6}\left[-\frac{2O}{T_4^2} + 0.5CD_4 + 0.5hD_4 - \frac{2S_1 n}{T_4^2} + 0.5\left(\theta_0 + \frac{\beta}{n}\right)\right] - \lambda_3 = 0$$

$$\frac{\partial L}{\partial \lambda_1} = -(T_2 - T_1), \quad \frac{\partial L}{\partial \lambda_2} = -(T_3 - T_2), \quad \frac{\partial L}{\partial \lambda_3} = -(T_4 - T_3)$$

$$T_1 = T_2 = T_3 = T_4 =$$

$$=\sqrt{\frac{12O+2(S_1n+2S_2n+2S_3n+S_4n)}{C(0.5D_1+D_2+D_3+0.5D_4)+h(0.5D_1+D_2+D_3+0.5D_4)+3\left(\theta_0+\dfrac{\beta}{n}\right)}}$$

The solution $\tilde{T}=(T_1,T_2,T_3,T_4)$ satisfies all inequality constraints. Let $T_1=T_2=$ $=T_3=T_4=\tilde{T}^*$ then the optimal value is:

$$\tilde{T}^*=\sqrt{\frac{12O+2(S_1n+2S_2n+2S_3n+S_4n)}{C(0.5D_1+D_2+D_3+0.5D_4)+h(0.5D_1+D_2+D_3+0.5D_4)+3\left(\theta_0+\dfrac{\beta}{n}\right)}}$$

Numerical example

Crisp model

The input parameters are O = 5000, β = 0,5, θ_0 = 7, S = 100, C = 500, h = 3000 and n = 3, we get

$$T=\sqrt{\frac{2(O+Sn)}{0.5CD+0.5hD+0.5\left(\theta_0+\dfrac{\beta}{n}\right)}}=0.0635459 \quad \text{and} \quad JC=333616.8$$

Fuzzy model

The input parameters are $\tilde{S}=(S_1,S_2,S_3,S_4)$, $\tilde{D}=(D_1,D_2,D_3,D_4)$, O = 5000, β = 0.5, θ_0 = 7, C = 500, h = 3000 and n = 3

S. no.	β	n	S = (79, 93, 110, 115) D = (1481, 1492, 1510, 1515)	S = (84, 94, 106, 116) D = (1485, 1496, 1506, 1511)
1	0.1	1	0.0623345	0.0623345
2	0.25	2	0.0629436	0.0629436
3	0.5	3	0.0635459	0.0635459
4	0.75	4	0.0641426	0.0641426
5	1	5	0.0647338	0.0647338

Machine learning and its methodology

Samir [22] discussed machine learning and logistic regression in PYTHON. The AI is a strategy to instruct programs that utilize information to produce calculations rather than expressly programming a calculation without any preparation. It is a field of software engineering that begins from the examination of man-made reasoning. It is firmly related to measurements and numerical improvement, which give techniques, hypotheses, and application spaces to the field. AI is utilized in different registering assignments where programming unequivocally rule-based calculations is infeasible.

Another supervised learning technique is logistic regression, which is a probabilistic classification model. It is primarily used to forecast a binary predictor. A logistic function is an extremely useful function that may take any value between –infinity and +infinity and output values between 0 and 1. As a result, it can be interpreted as a likelihood.

To test the integrated inventory model, we use PYTHON (3.8.5 version) code to predict the EOQ and total cost. Before starting the PYTHON code, we install the necessary packages like NumPy for arrays, pandas for data analysis and manipulation, matplotlib for data visualization and import MinMaxScaler, linear regression, r2_score, and train_test_split from scikit-learn library. We read the CSV file *final.csv* and using the describe function we find the mean, median, standard deviation, and percentiles. To test the model, we split the given dataset as training and testing to predict the EOQ and total cost. We use a training dataset to split the data. By using Linear Regression, we find the root mean square value and r2_score. Finally, we summarize the values of the linear regression by using the ordinary least squares method.

PYTHON code

```
# Importing basic libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from numpy import math
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
# Reading the CSV file and display first five rows from the dataset
dataset = pd.read_csv(r′ C:\Users\asus\Desktop\final.csv′)
dataset.head()
```

Table 1. First five rows of dataset

	item	holding cost	ordering cost	purchase cost	fixed rate	demand	coefficient	setup cost	inspection	t1	jc1
0	apple	1000	1625	2300	8	1630	0.5	100	5	0.039752	2.138260e+05
1	apple	6780	2596	1000	5	2400	0.6	110	4	0.025503	4.761857e+05
2	apple	5462	3478	1500	2	3400	0.8	120	6	0.026635	6.304602e+05
3	apple	8878	6789	2469	7	4500	0.9	105	2	0.023415	1.195625e+06
4	apple	3254	4522	4560	8	2450	0.4	100	8	0.033346	6.383918e+05

```
# Describe function to view statistical details
dataset.describe()
```

From the preceding table, we analyse the following result:

- The average mean of EOQ and the total cost is 0.026539 and 748144.1 respectively
- The standard deviation of EOQ and Total cost is 1.733188 and 203336.1 respectively.
- The minimum value of EOQ is 0.013035
- The Maximum value of total cost is 1195625.0

Table 2. Statistical analysis

	holding cost	ordering cost	purchase cost	fixed rate	demand	coefficient	setup cost	inspection	t1	jc1
count	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	100.000000	1.000000e+02
mean	5561.620000	4296.540000	3894.700000	6.420000	3254.290000	0.544000	118.850000	4.690000	0.026539	7.481441e+05
std	2055.259657	1599.561571	1330.701793	1.859619	961.567064	0.202669	28.022853	1.733188	0.007869	2.033361e+05
min	1000.000000	1456.000000	0.000000	2.000000	1450.000000	0.100000	75.000000	2.000000	0.013035	2.138260e+05
25%	3901.500000	3097.250000	2672.500000	5.000000	2477.750000	0.400000	95.000000	3.000000	0.021177	6.135388e+05
50%	5486.000000	4509.000000	4000.000000	6.000000	3425.000000	0.500000	110.000000	5.000000	0.025733	7.494338e+05
75%	6952.000000	5468.250000	4783.750000	8.000000	4020.000000	0.700000	145.000000	6.000000	0.031536	8.961018e+05
max	9876.000000	7896.000000	6530.000000	9.000000	4960.000000	1.000000	175.000000	8.000000	0.059718	1.195625e+06

```
# Scatter plot code for EOQ and ic
plt.scatter(dataset['eoq'],dataset['jc'],alpha=0.5)
plt.title('Scatter Plot of jc with EOQ')
plt.xlabel('EOQ')
plt.ylabel('jc')
plt.show()
```

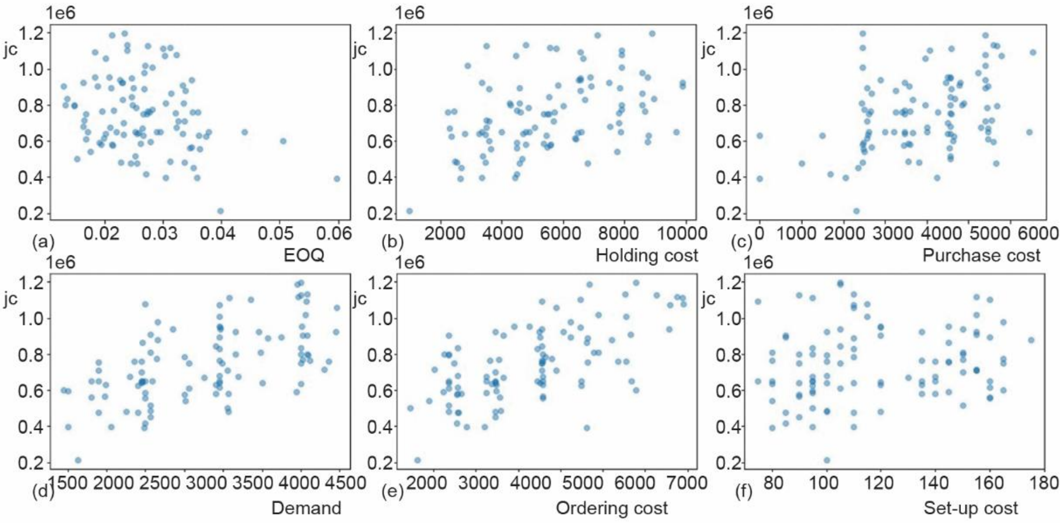

Figure 1. Scatter plot between EOQ and other costs; (a) scatter plot of jc with EOQ, (b) scatter plot of holding cost and jc, (c) scatter plot of purchase cost and jc, (d) scatter plot of demand and jc, (e) scatter plot ordering cost and jc, and (f) scatter plot of jc with EOQ

From fig. 1 it is noticed that all points are randomly scattered between EOQ and other costs.

```
# Logistic regression
dataset['apple_1']=np.where(dataset['item']=='apple',1,0)
dataset['mango_1']=np.where(dataset['item']=='mango',1,0)
dataset['papaya_1']=np.where(dataset['item']=='papaya',1,0)
dataset['cherry_1']=np.where(dataset['item']=='cherry',1,0)
dataset['guava_1']=np.where(dataset['item']=='guava',1,0)
dataset.drop(columns=['item'],axis=1,inplace=True)
```

Table 3. Splitting data set for testing

	holding cost	ordering cost	purchase cost	fixed rate	demand	coefficient	setup cost	inspection	t1	jc1	item_1	item_2	item_3	item_4	item_5
0	1000	1625	2300	8	1630	0.5	100	5	0.039752	2.138260e+05	1	0	0	0	0
1	6780	2596	1000	5	2400	0.6	110	4	0.025503	4.761857e+05	1	0	0	0	0
2	5462	3478	1500	2	3400	0.8	120	6	0.026635	6.304602e+05	1	0	0	0	0
3	8878	6789	2469	7	4500	0.9	105	2	0.023415	1.195625e+06	1	0	0	0	0
4	3254	4522	4560	8	2450	0.4	100	8	0.033346	6.383918e+05	1	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
95	8750	4256	4500	4	3456	0.6	145	5	0.020859	9.551753e+05	0	0	0	0	1
96	6588	4450	2600	5	2456	0.8	135	2	0.028925	6.527182e+05	0	0	0	0	1
97	7822	5632	4560	6	1900	0.5	145	3	0.032118	7.555953e+05	0	0	0	0	1
98	4566	5896	4890	8	2560	0.6	150	6	0.033511	8.112047e+05	0	0	0	0	1
99	3577	4563	2560	9	3550	0.7	155	8	0.032641	7.111294e+05	0	0	0	0	1

100 rows × 15 columns

```
dependent_variable='jc'
independent_variables=dataset.columns.tolist()
independent_variables.remove(dependent_variable)
independent_variables
X=dataset[independent_variables].values
y=dataset[dependent_variable].values
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=0)
scaler=MinMaxScaler()
X_train=scaler.fit_transform(X_train)
X_test=scaler.transform(X_test)
regressor=LinearRegression()
regressor.fit(X_train,y_train)
y_pred=regressor.predict(X_test)
math.sqrt(mean_squared_error(y_test,y_pred))
40705.09427694884
r2_score(y_test,y_pred)
0.9614306123056328
import statsmodels.api as sm
model1=sm.OLS(y_train,X_train).fit()
model1.summary()
OLS Regression Results
```

Dep. Variable:	y	R-squared:	0.981
Model:	OLS	Adj. R-squared:	0.978
Method:	Least Squares	F-statistic:	265.1
Date:	Tue, 29 Jun 2021	Prob (F-statistic):	1.30e-51
Time:	21:33:09	Log-Likelihood:	-930.14
No. Observations:	80	AIC:	1888.
Df Residuals:	66	BIC:	1922.
Df Model:	13		
Covariance Type:	nonrobust		

	coef	std err	t	$P>\|t\|$	[0.025	0.975]
x1	2.2e+05	2.35e+04	9.352	0.000	1.73e+05	2.67e+05
x2	5.943e+05	2.72e+04	21.827	0.000	5.4e+05	6.49e+05
x3	1.789e+05	2.47e+04	7.252	0.000	1.3e+05	2.28e+05
x4	-2.564e+04	1.29e+04	-1.994	0.050	-5.13e+04	34.535
x5	3.025e+05	2.46e+04	12.320	0.000	2.53e+05	3.51e+05
x6	1.3e+04	1.6e+04	0.811	0.420	-1.9e+04	4.5e+04
x7	1.034e+05	2.75e+04	3.762	0.000	4.86e+04	1.58e+05
x8	4.658e+04	1.25e+04	3.720	0.000	2.16e+04	7.16e+04
x9	-3.055e+05	6.24e+04	-4.895	0.000	-4.3e+05	-1.81e+05
x10	1.75e+05	4.08e+04	4.287	0.000	9.35e+04	2.56e+05
x11	1.686e+05	4.15e+04	4.065	0.000	8.58e+04	2.51e+05
x12	1.217e+05	4.52e+04	2.693	0.009	3.15e+04	2.12e+05
x13	1.672e+05	4.07e+04	4.110	0.000	8.59e+04	2.48e+05
x14	1.22e+05	4.42e+04	2.759	0.007	3.37e+04	2.1e+05

Omnibus:	14.766	**Durbin-Watson:**	1.936
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	22.460
Skew:	0.741	**Prob(JB):**	1.33e-05
Kurtosis:	5.132	**Cond. No.**	52.6

The root mean squared value is 407050.942, R^2 score is 0.96143 and R-Squared value is 0.981, indicating that the dataset is 98% acurate.

Conclusion

An inventory model for deteriorating rate items is studied in order to minimise total cost and maximise profit, and solved for both crisp and fuzzy. Defuzzification is done by graded mean integration. The fuzzy inventory model is solved by the non-linear Lagrangian method. Moreover, we use PYTHON code to predict the values of EOQ and the total cost. As a result, we get 98% accuracy for the given dataset *i.e.*, when the value of EOQ increases total cost is also increased.

Funding Statement

The authors extend their appreciation to the Deanship of Post Graduate and Scientific Research at Dar Al Uloom University for funding this work.

References

[1] Ghare, P. M., Schrader, G. P., A Model for Exponentially Decaying Inventory, *Journal of Industrial Engineering*, *14* (1963), pp. 238-243

[2] Covert, R. P., Philip, G. C., An EOQ Model for Item with Weibull Distribution Deterioration, *AIIE Transactions*, *5* (1973), 4, pp. 323-326

[3] Dave, U., Patel, L. K., Policy Inventory Model for Deteriorating Items with Time Proportional Demand, *Journal of Operational Research Society*, *32* (1981), 2, pp. 137-142

[4] Misra, R. B., Optimum Production Lot-Size Model for a System with Deteriorating Inventory, *International Journal of Production Research*, *13* (1975), 5, pp. 495-505

[5] Sachan, R. S., On (T, S_i) Inventory Policy Model for Deteriorating Items with Time Proportional Demand, *Journal of Operational Research Society*, *35* (1984), 11, pp. 1013-1019

[6] Dye, C. Y., *et al*., Determining Optimal Selling Price and Lot Size with a Varying Rate of Deterioration and Exponential Partial Backlogging, *European Journal of Operational Research*, *181* (2007), 2, pp. 668-678

[7] Salameh, M. K., Jaber, M. Y., Economic Production Quantity Model for Items with Imperfect Quality, *International Journal of Production Economics*, *64* (2000), 1-3, pp. 59-64

[8] Ben-Daya, M., *et al*., Integrated Inventory Control and Inspection Policies with Deterministic Demand, *Computers and Operations Research*, *33* (2006), 6, pp. 1625-1638
[9] Harris, F. W., How Many Parts to Make at Once, Factory, *The Magazine of Management*, *10* (1913), 2, pp. 135-136
[10] Zadeh, L. A., Fuzzy Sets, *Information and Control*, *8* (1965), pp. 338-353
[11] Zimmermann, H. J., Description and Optimization of Fuzzy Systems, *International Journal of General Systems*, *2* (1976), 1, pp. 209-215
[12] Kaufmann, M., Gupta, M., *Introduction to Fuzzy Arithmetic Theory and Applications*, Van Nostrand, Reinhold, New York, USA, 1985
[13] Chen, S. H., Hsieh C. H., Graded Mean Integration Representation of Generalized Fuzzy Number, *Journal of Chinese Fuzzy Systems*, *5* (1999), 2, pp. 1-7
[14] Kalaiarasi, K., *et al*., Optimizing EOQ Using Geometric Programming with Varying Fuzzy Numbers by Applying PYTHON, *Journal of Critical Reviews*, *7* (2020), pp. 596-603
[15] Siva Jyothi, P. N., Rohita, Y., A Review on PYTHON for Data Science, Machine Learning and IOT, *International Journal of Scientific & Engineering Research*, *10* (2019), 12, pp. 851-858
[16] Chen, S. H., Operations on Fuzzy Numbers with Function Principle, *Tamkang Journal of Management Science*, *6* (1986), 1, pp. 13-25
[17] Halima, K. A., *et al*., Fuzzy Production Planning Models for an Unreliable Production System with Fuzzy Production Rate and Stochastic/Fuzzy Demand Rate, *International Journal of Industrial Engineering Computations*, *2* (2011), 1, pp. 179-192
[18] Khakzad, A., Gholamian, M. R., The Effect of Inspection on Deterioration Rate: An Inventory Model for Deteriorating Items with Advanced Payment, *Journal of Cleaner Production*, *254* (2020), May, pp. 120-117
[19] Hamdy, A. T., *Operations Research Introduction*, 8th ed., Pearson Publisher, London, UK, 2008
[20] Alsaraireh, A. A., *et al*., Comparing Study Between Simplex Method and Lagrange Method in a Linear Programming Problem, *Italian Journal of Pure and Applied Mathematics, 4* (2019), Aug., pp. 934-943
[21] Kalaiarasi, K., *et al*., Optimization of Fuzzy Economic Production Quantity Inventory Control, *Journal of Advanced Research in Dynamical and Control System*, *11* (2019), Speciall Issue, pp. 241-248
[22] Samir, M., *Mastering PYTHON for Data Science*, Packt Publishing, Mumbai, India, 2015

Paper submitted: June 15, 2021
Paper revised: July 5, 2021
Paper accepted: July 18, 2021