GLOBAL ACADEMY OF MATHEMATICAL AND ECONOMIC SCIENCES

March 2022

GUIDE TO PORTFOLIO PROJECT

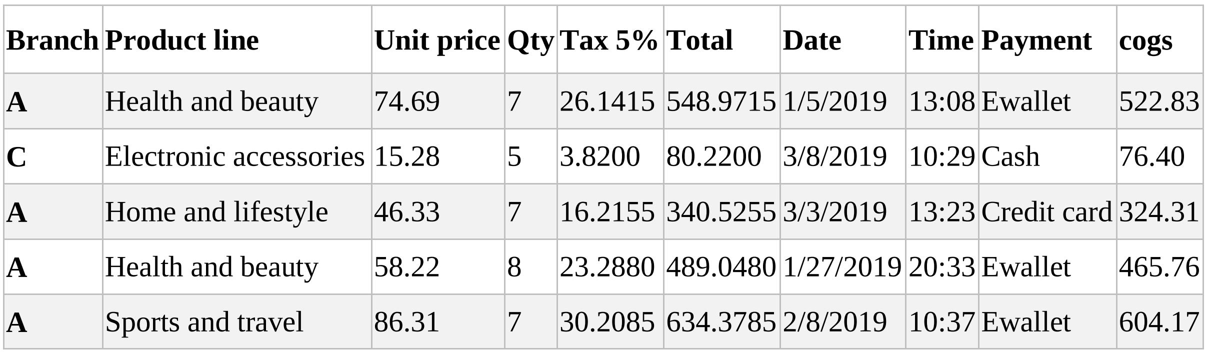
Probabilistic Sales Forecasting using Discrete Distribution Models

Project Creator/Supervisor: Zion Pibowei

Course: Probabilistic Modelling

This guide is a run-through of the approach and techniques expected in fulfilling the technical objectives of the portfolio project.

Here is a glance at the first 5 rows of the data. The columns City, Customer type, Gender, gross margin percentage, gross income, and Rating have been removed for the sake of brevity.



Branch	Product line	Unit price	Qty	Tax 5%	Total	Date	Time	Payment	cogs
A	Health and beauty	74.69	7	26.1415	548.9715	1/5/2019	13:08	Ewallet	522.83
C	Electronic accessories	15.28	5	3.8200	80.2200	3/8/2019	10:29	Cash	76.40
A	Home and lifestyle	46.33	7	16.2155	340.5255	3/3/2019	13:23	Credit card	324.31
A	Health and beauty	58.22	8	23.2880	489.0480	1/27/2019	20:33	Ewallet	465.76
A	Sports and travel	86.31	7	30.2085	634.3785	2/8/2019	10:37	Ewallet	604.17

The first thing you should do is convert the Date field to a datetime data type after replacing “2019” with “2022”. Yes, we want to treat the data as historical records for Q1 2022 and make forecasts for Q2.

Now, let’s see how to approach the objectives, one by one:

1. Binomial Experiments:

(a) Use a binomial experiment to simulate the number of times sales will be recorded in Branch A in Q2 daily. Repeat the simulation for branches B and C, and aggregate the results.

Notes:

A binomial experiment is a sequence of trials where each trial has 2 possible outcomes and can only result in one of them. The results of a binomial experiment follow the binomial probability distribution, & the probability of obtaining any of the outcomes of the binomial distribution is given by the binomial probability mass function (PMF) as follows:

$$P(X = k) = \binom{n}{k} p^k (1 - p)^{n-k}$$

Where X is the random variable of interest, k is a specific value of X at an instance, n is the fixed number of trials in the experiment and p is the probability of realising each successful outcome. This probability is known or calculated beforehand.

From the objective, we are interested in predicting successful outcomes, so we need to simulate a binomial distribution $X \sim B(n,p)$, then generate a draw (random outcome k) from that distribution. To simulate a binomial distribution in Python, use numpy's random module and run binomial(n,p) which simulates the $B(n,p)$ distribution and returns a random outcome. Since objective 1(a) is focused on simulating number of times sales were recorded each day, rather than number of quantity sold, the number of trials n will be total number of times sales were made each day. So we are running different simulations for each of the days. To get random draws from the binomial distribution for each day, here's a possible algorithm:

Algorithm 1.1

- Get array of total number of times sales were recorded each day:
  $N = \{n_i \text{ for } i \in D\}$
- Get array of daily probabilities of success for branch A:
  $$P(A) = \left\{\frac{n(A_i)}{n_i} \text{for } i \in D\right\}$$
- Run element-wise binomial simulation for elements in $\{N, P(A)\}$:
  $\{B(n,p) \text{ for } (n,p) \in \{N \times P(A)\}\}$

This results in an array containing outcomes simulated for each day. Daily results will vary for each implementation of the algorithm. To make results a bit more stable we can generate multiple draws (e.g., $M = 1000$) for each day and get the daily mean. This requires a slight change to step 3 in Algorithm 1.1, as follows:

$$\left\{\frac{1}{M}\sum B(n,p,M) \text{ for } (n,p) \in \{N \times P(A)\}\right\}$$

A binomial distribution doesn't take account of time so we can only simulate possible outcomes that can occur but not for a specific time in the future. A good way to make realistic projections for each day in Q2 *could be* to combine the simulation over the entire data with the simulation for that specific day. This brings us to Algorithm 1.2 as follows:

Algorithm 1.2

- Get the following prior information for entire data:
  - Total number of times n_E sales were recorded throughout entire sales history
  - Total number of times n_A sales were recorded in Branch A throughout history
  - Probability of success for Branch A for the entire history: $p_A = n_A / n_E$
- Get prior information for each day:
  - Total number of times sales were recorded each day:

    $N = \{n_i \text{ for } i \in D\}$
  - Daily probabilities of success for branch A:

    $$P(A) = \left\{ \frac{n(A_i)}{n_i} \text{for } i \in D \right\}$$
- For each element in $\{N, P(A)\}$, generate a draw from binomial simulation for that instance, add it to the outcome from binomial simulation of entire data, & subtract historical number of sales events in branch A:

$$\left\{ B\left(n_E, p_E\right) + B(n, p) - n_A \text{ for } (n, p) \in \{N \times P(A)\} \right\}$$

Here's the intuition behind the algorithm: Say there are 90 days in the data and we want to simulate for the 91st day. We can simulate over 90 days, add it to a single day simulation, so that the resulting outcome is the total for 91 days. Then we subtract the total historical number of outcomes for 90 days. That leaves us with the outcome for 1 day, namely the outcome of the 91st day of interest. We can stabilise the results by taking the mean of multiple draws from the 90-day simulation, using $\frac{1}{M} \sum B\left(n_E, p_E, M\right)$, and round off to the next integer since outcomes must be discrete.

Objective 1(b): Simulate the quantity of products that will be sold in each of the branches in Q2 daily.

This is quite similar to objective 1a), only difference being that we are taking account of each quantity sold and where they were sold, rather than just the number of times bulk sales were recorded.