Code Appendix or Variable description:

Main loop 1 and User inputs:

c = Current total inventory remaining.

c0 = Initial inventory, before or at start of booking period

customer_data = List to store customer data

customer_id = 1 = Serial number given to each customer as they approach irrespective of bought or not. This is new even if a customer approaches more than one time.

customers_per_price = number of customers approaching per price

demand_history = list to store demand values with respect to price for plotting price vs demand for Phase 1.

delta = (δ) parameter used in exploration for narrowing price intervals near optimal price.

df = data frame for demand generated in 1 subphase in Exploration (Phase 1).

df_1 = data frame for complete Exploration (Phase 1).

df_1 = dataframe to store Phase 1 demand simulation data

df_1_type_1 = dataframe to store Phase 1 demand simulation data only for customer type 1.

df_1_type_2 = dataframe to store Phase 1 demand simulation data only for customer type 2.

K = Total subphases in phase 1 (K x t = P1)

ki = 1 Subphase in Phase 1 or 1 element in K range

num_steps = Total number of Price points wanted

p = Current price list for type 1 (Reserve Customers), initially created after inputs of Highest and Lowest price taken form user.

p_1_initial = Initial price list for customer type 1, this does not change.

p_2_initial = Initial price list for customer type 2, this does not change.

p2 = Current price list for type 2 (Preemptive Customers), initially created after inputs of Highest and Lowest price taken form user.

p_low = lower price limit when we narrow price intervals towards optimal price.

p_opt_1 = p, *optimal price for 1st type (Reserve Customers) in Exploration.*

p_opt_2 = p, optimal price for 2nd type (Preemptive Customers) in Exploration.

p_opt_type1 = list for collecting optimal prices of type 1 customer in phase 1

p_opt_type2 = list for collecting optimal prices of type 2 customer in phase 1

p_upp = upper price limit when we narrow price intervals towards optimal price.

pt = price for a particular instance, pt is in range p.

P1 = total booking time of phase 1.

P2 = total percbooking time of phase 2, P1+P2 = T, a phase can have many subphases.

T = current total booking tim remaining

T0 = Initial booking peroid

t = time instances to divide booking time.

total_customers = 2 x customers_per_price x num_steps (total price points), total customers approaching for a subphase ki in Phase 1. 2 is for 2 customer types.

unit_T = unit of booking peroid (Usually days)

xt = Unit of Booking time T one instance has.

z = dual variable at beginning, can be zero at 1st instance.

z_opt = z*, optimal dual variable for both types (Reserved and Preemptive) customers.

Exploration function (Phase 1):

customer_data = List to store demand simulation data as customer approaches

customer_id = an serial number order number given to all customers starting from 1 irrespective of bought or not.

demand_probability_1 = demand probability calculation for customer type 1 after demand simulation and recording.

demand_probability_2 = demand probability calculation for customer type 2 after demand simulation and recording.

demand_history = list to store demand values with respect to price for plotting price vs demand for Phase 1.

df = dataframe inside exploration funciton consisting data for demand simulation for that iterating subphase in phase 1 (exploration)

new_price_list_1 = New price list for customer type 1 found after narrowing price intervals via delta

new_price_list_2 = New price list for customer type 2 found after narrowing price intervals via delta

p_low_1 = new lower price interval found after narrowing with respect to p_opt_1

p_low_2 = new lower price interval found after narrowing with respect to p_opt_2

p_opt_1 = Optimal price for customer type 1

p_opt_2 = Optimal price for customer type 2

p_upp_1 = new upper price interval found after narrowing with respect to p_opt_1

p_upp_2 = new upper price interval found after narrowing with respect to p_opt_2

revenue_per_price_1 = revenue per price for customer type 1

revenue_per_price_2 = revenue per price for customer type 2

total_revenue = revenue for both prices. here, revenue is normal revenue i.e. price x units sold.

total_revenue_1 = total revenue for customer type 1, here, revenue is normal revenue i.e. price x units sold.

total_revenue_2 = total revenue for customer type 2, here, revenue is normal revenue i.e. price x units sold.

units_sold = inventory sold in one subphase of phase one.

z_opt = Combined dual variable value for both customers types

z_opt_1 = Dual variable, Optimal unit value of inventory obtained from demand data of customer type 1

z_opt_2 = Dual variable, Optimal unit value of inventory obtained from demand data of customer type 2

Main loop 2:

df_2 = dataframe to store Phase 2 demand simulation data.

df_2_type_1 = dataframe to store Phase 2 demand simulation data for customer type 1.

df_2_type_2 = dataframe to store Phase 2 demand simulation data for customer type 2.

safe_T = Safe booking time limit, T > safe_T ensures booking peroid does not become negative while conducting exploitation's last iteration

safe_c = Safe Inventory, to avoid inventory going (-ve) negative

ti = iteration variale which will be used to see what iteration we are in while doing exploitation phase.

Exploitation function (Phase 2):

alpha_1 = Alpha (bound length for a optimal price) for customer type 1

alpha_2 = Alpha (bound length for a optimal price) for customer type 2

D_lower_1 = demand probability calculated from df_temp_lower for customer type 1

D_lower_2 = demand probability calculated from df_temp_lower for customer type 2

D_upper_1 = demand probability calculated from df_temp_upper for customer type 1

D_upper_2 = demand probability calculated from df_temp_upper for customer type 2

demand_estimate_1 = Estimated demand for optimal price for customer type 1 via *llightgbm_buying_probability* function.

demand_estimate_2 = Estimated demand for optimal price for customer type 2 via *llightgbm_buying_probability* function.

df_exploitation = data frame to store demand simulation for complete 1 iteration of exploitation.

df_temp_lower = = temporary data frame in exploitation function for storing demand simulation for half instance of lower bound on both customer types.

df_temp_upper = temporary data frame in exploitation function for storing demand simulation for half instance of upper bound on both customer types.

p_opt_1_high = higher bound for Optimal price for customer type 1, p_opt_1 + alpha_1

p_opt_1_low = Lower bound for Optimal price for customer type 1, p_opt_1 - alpha_1

p_opt_2_high = higher bound for Optimal price for customer type 2, p_opt_2 + alpha_2

p_opt_2_low = Lower bound for Optimal price for customer type 2, p_opt_2 - alpha_2

theta_1 = Time allocation metric for price bounds for customer type 1.

theta_1_lower_time = time allocated to lower bound of price for customer type 1

theta_1_upper_time = time allocated to upper bound of price for customer type 1

theta_2 = = Time allocation metric for price bounds for customer type 2.

theta_2_lower_time = time allocated to lower bound of price for customer type 2

theta_2_upper_time = time allocated to upper bound of price for customer type 2

total_estimated_demand = total of demand_estimate_1 and demand_estimate_2

total_future_demand = Total future demand estimated based on current demand according to current optimal prices.

*Preemptive means Preemptable but since it is used from beginning and is throughout the code, it is kept as Preemptive.