Stochastic Inventory Simulator

(Newsvendor Problem)

# Introduction

This project implements a simulation of the Newsvendor Problem, a classic inventory management model. It helps determine the optimal production quantity for a single-period product with uncertain demand. The simulator uses Monte Carlo simulation and optimization techniques to find the production quantity that maximizes expected profit.

# Purpose

The purpose of this project is to:

* Provide a tool for understanding and exploring the Newsvendor problem.
* Demonstrate the application of probabilistic methods and optimization in inventory management.
* Allow users to experiment with different demand and observe their impact on optimal production quantities.

# Features

* Monte Carlo Simulation: Simulates demand using normal distribution
* Profit Calculation: Calculates profit based on production quantity, demand, selling price, cost and salvage value
* Optimization: Uses scipy.optimize\_scalar to find the optimal production quantity.
* Graphical User Interface (GUI): Provides a user-friendly interface using Tkinter
* Configurable Parameters: Allows user to input mean demand, standard deviation and production quantity

# Technology Used

* Python
* NumPy (for numerical operations)
* SciPy (for optimization)
* Tkinter (for GUI)

# Installation and Usage

1. Prerequites:
   * Python 3.x
   * Install required libraries: pip install numpy scipy
2. Running the application:
   * Make sure your current directory in your terminal is stochasticInventorySimulator before running the python script:



1. Using the GUI:
   * Enter the mean demand and standard deviation (default values are 1000 and 200 respectively)
   * Optionally, enter a specifiy production quantity
   * Click the “Run Simulation” button
   * The results (average profit or optimal quantity and maximum profit) will be displayed in the result label.

# Code Structure

* Main script
  + Contains the TKinter GUI code
  + Imports the simulation and optimization functions
  + Handles user input and displays results.
* Simulation function (newsvendor\_simulation(mean, stddev, production\_quantity, num\_trials=1000))
  + Simulates the Newsvendor problem using normal distribution
  + Calculates sales, unsold quantity, and profit for each trial
  + Returns the average profit over all trials
* Optimization function
  + Finds the optimal production quantity using scipy.optimize.minimize\_scalar
  + minimize\_scalar finds the minimum by default. Since we want to maximize profit, we define an inner function that takes quantity as input and return the negative of the profit. This inner functions calls the newsvendor\_simulation(…) to get the average profit for that quantity and returns the negative of the profit.
  + We’ll call minimize\_scalar with the function to minimize and bounds of our search
  + We extract the result of the main function:
    - optimal\_quantity = int(result.x) : The oprimal quantity is store in result.x. We convert it to an integer
    - max\_profit=-result.fun: The minimum value of the function(which is the negative of the maximum profit) is store in result.fun. We invert it to get the actual maximum profit.

# Optimization Details

* The optimization uses scipy.optimize.minimize\_scalar with the ‘bounded’ method
* The function returns the minimized value therefore we need to transform the result to wnsure that we are maximizing the actual profit.
* The search for the optimal quantity is done within the used defined bounds.

# Potential Enhancements

* Add support for other demand distributions (e.g. Poisson, uniform)
* Implement more advanced optimization algorithms
* Add features to visualize simulation results (e.g. histrograms of demand and profit)
* Add more user configurable parameters, such as the cost per unit, and the selling price
* Implement data persistence, so that previous runs can be saved and reloaded
* Create a web based version of the application