In this assignment, you solve problems for Stochastic Linear Programming and Quadratic Programming. You need to

(1) Formulate the Stochastic Programming model and solve it with Python and Gurobi.

(2) Formulate the Quadratic Programming model and solve it with Python and Gurobi.

1. **Submission Instructions**

Submit a PDF file describing stochastic programming (SP) and quadratic programming (QP) models and the solutions to the problem instance. Also, submit a program (a Python script or a Jupyter notebook) using Gurobi to solve the problem instance. In the formulation, clearly define decision variables and state the objective function and constraints. For the problem instance, report the values of the objective and decision variables.

2. **Problem**

Question 1: You are a production manager at SweetFlow Syrups, a company specializing in producing a variety of flavored syrups. SweetFlow produces three types of syrups: vanilla (V), maple (M) and cherry (C). The company faces significant demand variability, especially due to seasonal changes and market trends. To manage this uncertainty effectively, you decide to use stochastic programming for production planning.

Your task is to develop a monthly production plan to maximize the total expected profit over the next month, considering the uncertainties in demand. You have historical data that suggests the demand and cost scenarios can be modeled with discrete distributions.

Storage: Notably, any syrup produced in excess of the demand within the month must be disposed of, as the company's storage capabilities do not permit carrying over inventory to the next month.

Raw Material Cost (per liter): Raw Syrup ($0.5/liter) and Flavor Concentrates ($2/liter for vanilla and maple, $2.5/liter for cherry).

Pricing (per liter): $6/liter for vanilla, $7/liter for maple and $8/liter for cherry.

Demand Scenarios: The demand of vanilla syrup can be high (900 liters with 40% probability) and low (500 liters with 60% probability). The demand of maple syrup can be high (600 liters with 50% probability) and low (300 liters with 50% probability). The demand of cherry syrup can be high (1100 liters with 70% probability) and low (700 liters with 30% probability). The probability of demands for different syrups is independent.

Raw Syrup Production (per month): Due to equipment limitations and labor constraints, the maximum production of raw syrup is 1800 liters per month.

Minimum Production Requirement: SweetFlow has contracts with suppliers that require at least 300 litres production amount for each syrup to maintain their business relationships.

Formulate the stochastic programming problem and then solve it using Python and Gurobi.

Question 2: Non-Negative Least Squares (NNLS)

Background:

The Least Squares problem is a fundamental optimization problem in mathematics, statistics, and engineering, primarily used for fitting a model to data. It finds the best-fitting curve or line to a set of points by minimizing the sum of the squares of the differences between the observed values and those predicted by the model. NNLS is a variant of the Least Squares problem that adds a non-negativity constraint on the solution. This constraint is crucial in many real-world applications where the quantities being estimated are inherently non-negative, such as concentrations of substances, pixel values in images, or amounts of materials.

Problem Statement:

You are provided with a set of measurements obtained from a sensor array that captures information about signal $\boldsymbol{x}$. Each signal has been transformed through a known measurement process $\widetilde{\boldsymbol{b}} = \boldsymbol{A}\boldsymbol{x}$, resulting in a measurement $\boldsymbol{b}$ that are lower in dimensionality than the original signal $\boldsymbol{x}$. Your task is to reconstruct the original signal $\boldsymbol{x}$ using NNLS optimization, ensuring all signal values are non-negative.

Dataset:

Let's define the measurement matrix $\boldsymbol{A}$ and the measured projections $\boldsymbol{b}$ with numerical details:

$$\boldsymbol{A} = \left\{\begin{matrix} 1.0 & 0.5 & 0.3 & 0.2 & 0.1 & 0.7 & 0.2 & 0.1 \\ 0.5 & 1.0 & 0.4 & 0.3 & 0.2 & 0.1 & 0.3 & 0.9 \\ 0.4 & 0.4 & 1.0 & 0.8 & 0.6 & 0.4 & 0.2 & 0.1 \\ 0.3 & 0.6 & 0.8 & 1.0 & 0.8 & 0.6 & 0.4 & 0.2 \\ 0.2 & 0.2 & 0.5 & 0.8 & 1.0 & 0.8 & 0.6 & 0.4 \end{matrix}\right\}, \boldsymbol{b} = \begin{bmatrix} 0.3 \\ 1.0 \\ 0.8 \\ 0.7 \\ 0.5 \end{bmatrix}$$

Question:

Formulate the quadratic problem to reconstruct the original signal $\boldsymbol{x}$ and then solve it using Python and Gurobi.