# Machine learning – Lab 2

# Exercise 1: Excel Solver

1. Use the attached ***solver.xlsx*** excel file and use Solver to solve the linear programming problem described at the following link.

<https://www.youtube.com/watch?v=6xa1x_Iqjzg>

Note: In Excel 2016, you may have to choose File -> Options -> Add Ins -> Solver -> Go -> Solver

-> Ok

You can find solve under the ‘Data’.

Save the updated file.

Note: For the following exercises on your own pc/laptop, you may install Anaconda first.

<https://www.anaconda.com/distribution/>

# Exercise 2: Linear Programming for Diet Optimization

## Objective

In this lab, you will use linear programming to design an affordable diet plan that meets basic nutritional needs while minimizing cost. You will determine the optimal combination of food items (Chicken, Rice, and Broccoli) that satisfies daily calorie, protein, and fat requirements at the lowest possible cost.

You will be using Scipy python library for this exercise. **SciPy** is an open-source Python library used for scientific and technical computing, providing efficient numerical methods for optimization, integration, linear algebra, and more. It builds on NumPy and includes modules like scipy.optimize for optimization, scipy.stats for statistics, and scipy.signal for signal processing, making it a powerful tool for engineers, scientists, and data analysts.

## Problem Statement

You have three food items available for your diet:   
- Chicken: High in protein but relatively expensive.  
- Rice: Cheap and provides carbohydrates but lacks protein.  
- Broccoli: A good source of fiber and vitamins.  
  
Your goal is to minimize the total cost while meeting daily nutrition requirements.

## Food Nutrition Information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Food Item | Cost per Serving ($) | Calories per Serving | Protein (g per Serving) | Fat (g per Serving) |
| Chicken | $2.50 | 250 | 30 | 8 |
| Rice | $0.80 | 130 | 3 | 0.5 |
| Broccoli | $1.50 | 50 | 4 | 0.2 |

## Constraints

To maintain a healthy diet, you must meet the following minimum daily nutrition requirements:

- Calories: At least 2000 kcal per day  
- Protein: At least 50g per day  
- Fat: At least 10g per day

## Mathematical Formulation

Objective Function:

Minimize cost:  
Minimize: 2.5x + 0.8y + 1.5z

Constraints:

1. Calories Constraint:

250x + 130y + 50z ≥ 2000

2. Protein Constraint:

30x + 3y + 4z ≥ 50

3. Fat Constraint:

8x + 0.5y + 0.2z ≥ 10

4. Non-negativity Constraint: x, y, z ≥ 0

## Python Implementation

Upload and run the **DietPlanLP** jupyter notebook.

**Note:** If you get an error when running the notebook that says, ‘Unkown Solver highs’ you may need to upgrade the scipy library. Create a code cell above the current cell and Run the following command and restart Jupyter notebook.

!pip install --upgrade --user scipy

## Extension Task

Modify the program to include an additional constraint for Vitamin C intake:  
- Broccoli provides 50 mg of Vitamin C per serving  
- You must consume at least 30 mg of Vitamin C per day  
Update the constraints and observe how the diet changes.

**Note:** The TODO: comments give hints on how to update the code.

## Writing Task

After running the optimization, write a short reflection answering the following:  
1. Before adding Vitamin C: What was the optimal diet and cost?  
2. After adding Vitamin C: How did the results change? Did the cost increase?  
3. What does this tell you about trade-offs in diet planning?

**Note:** You can type the answers by adding a text cell at the end of the notebook file.

**Download and save the updated Notebook.**

## Additional Resources

For more details on `scipy.optimize.linprog`, refer to the official SciPy documentation:

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.linprog.html>

# Exercise 3: Optimizing Delivery Routes using Genetic Algorithm

Note: You may have to run the following command to install the DEAP library from notebook.

## !pip install deap

## Objective

In this lab assessment, you will use a Genetic Algorithm (GA) to optimize a delivery route for a logistics company. The goal is to minimize the total travel distance while ensuring all locations are visited efficiently.   
  
You will use DEAP (Distributed Evolutionary Algorithms in Python) to implement and run the genetic algorithm.

## Problem Statement

A delivery company wants to find the most efficient route to visit 6 locations, starting and ending at a warehouse. This problem is similar to the Traveling Salesman Problem (TSP), where the goal is to find the shortest possible route visiting all locations exactly once.  
  
Your task: Implement a Genetic Algorithm in Python using the `DEAP` library to optimize the delivery route.

## Instructions

1. Run the provided Jupyter notebook to find the optimal delivery route using GA.  
2. Understand how selection, crossover, and mutation influence optimization.  
3. Modify the fitness function to introduce time constraints (additional task).  
4. Re-run the modified genetic algorithm and analyze the impact.

## Python Implementation

Run the given DeliveryGA notebook. Follow the TODO comments to complete the additional tasks.

## Additional Task: Modify the Fitness Function

Currently, the fitness function only minimizes the total travel distance.  
  
Your Task: Modify it to consider time constraints.  
  
Steps:  
1. Define a list of preferred delivery times for each location (e.g., `[2, 4, 1, 3, 5]`).  
2. Modify the fitness function to penalize routes that visit locations after their preferred time slot.  
3. Re-run the modified genetic algorithm and analyze how the best route changes.

## Analysis Questions

After modifying and running the program, answer the following:  
1. Before modifying the fitness function, what was the optimal route and distance?  
2. After including time constraints, how did the route change? Did the cost increase?  
3. What happens when you increase the **number of generations**? What happens when you update the **mutation rate**, or change the **size of the population**? Briefly explain the observations.

Write the answers to the above questions in the text cell at the bottom of the notebook.

Download and save the updated Notebook.

## Submission

Zip the two notebooks and the excel file to a single zip file. The file name should be your registration number. Upload it to the courseweb link.