Project Ideas

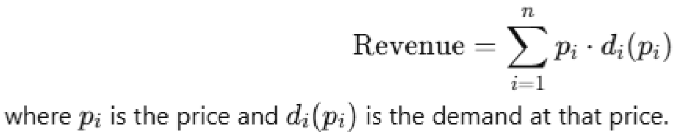
1. Pricing Optimization:

Objective:
Develop a pricing model that optimizes revenue or profit while adhering to specific constraints, such as maximum production capacity or minimum acceptable profit margins.

Tasks:

1. **Define the Pricing Model:**

   ➢ Design a pricing model with an objective function (e.g., maximize revenue):



$$\text{Revenue} = \sum_{i=1}^{n} p_i \cdot d_i(p_i)$$

where p_i is the price and $d_i(p_i)$ is the demand at that price.

   ➢ Add constraints like maximum price or demand.

   ➢ Solve the optimization problem using **CVXPY**.

   - Use a convex optimization approach to maximize revenue or minimize costs.

2. **Determine Convexity:**
   - Check whether the revenue function is convex using **Hessian matrix** or CVXPY features:
     1. If the function is convex, ***justify this mathematically***.
     2. If not convex, explain why.

3. **Modify the Model (Non-Convexity):**
   - Introduce ***modifications*** to ***make the model non-convex***:
     1. For example, add a non-linear term in the demand function to make it more complex.

4. **Restore Convexity:**
   - ***Modify the model again*** (e.g., remove the non-linear term or simplify the function) to restore convexity.

5. **Visualization:**
   - Use **Matplotlib** or **Seaborn** to plot the relationship between price and revenue in the three cases:
     1. Original (Convex).
     2. Non-Convex.
     3. Restored Convex.

Outcomes:

- Python script implementing the optimization problem.
- A presentation or report summarizing:
  - The problem setup (objective and constraints).
  - Key insights from the analysis.
  - Visualizations showing pricing vs. revenue trends.

2. Financial Data Analysis:

Objective:
Analyze a dataset of *financial transactions* or *stock prices* to identify patterns, optimize investment portfolios, or improve financial performance.

Tasks:

1. **Use a simple real dataset.**

2. **Setup a Financial Model:**
   - Define an objective function (e.g., maximize returns, minimize risk, …).
   - Define the constraints.
   - Formulate the Problem.
   - Solve the optimization problem using **CVXPY**.
   - … .

3. **Determine Convexity:**
   - Check whether the objective function is convex using properties of convex functions (e.g., variance is always convex, ….) or CVXPY features:
     1. If the function is convex, ***justify this mathematically***.
     2. If not convex, explain why.

4. **Modify the Model (Non-Convexity):**
   - Introduce non-linear or varying components to modify the portfolio and make it non-convex:
     1. For example, add non-linear constraints to stock allocations.

5. **Restore Convexity:**
   - Simplify the model (e.g., remove non-linear constraints or use a convex approximation).

6. **Visualization:**

    - Use Matplotlib or Seaborn to visualize financial trends, portfolio performance, or risk-return tradeoffs.
    - Use graphs to display the relationship between risk and return in the three cases:
        1. Original (Convex).
        2. Non-Convex.
        3. Restored Convex.

Outcomes:

- Python script implementing the optimization problem.
- A presentation or report summarizing:
    - The problem setup (objective and constraints).
    - Key findings from the analysis.
    - Visualizations (e.g., portfolio allocation, stock trends).

3. Students' Exam Scores Analysis:

Objective:

Analyze student exam scores, verify the convexity of the score distribution, modify the distribution to make it non-convex, and then restore convexity.

Tasks:

1. **Analyze the Scores and (Optimization):**
   - Analyze the student score data and its distribution:
     - Example: A simple distribution with mean and standard deviation.
     - ***Objective function:*** Minimize variance among scores for fairness or the goal could be to improve grades or reduce differences between students' grades (balance students' grades).
     - ***Constraints:*** Impose some constraints such as minimum grades or maximum grades in each subject.
     - Solve the problem using CVXPY.

2. **Determine Convexity:**
   - Check the convexity of the distribution:
     - Use CVXPY to analyze the objective function.
     - Justify mathematically whether the distribution function is convex or not.

3. **Modify the Distribution (Non-Convexity):**
   - Introduce modifications to the data to make it non-convex:
     - Example: Add outliers or change the grades in a non-linear manner.

4. **Restore Convexity:**
   - Use techniques like:
     - Removing outliers.
     - Adding a constraint to the distribution to make it more organized.

5. **Visualization:**
    - Display the distribution before and after modification using graphs.
    - Compare the three cases:
        - Original (Convex).
        - Modified (Non-Convex).
        - Restored (Convex).

Outcomes:

- Python script implementing the analysis and optimization.
- A presentation or report summarizing:
    - The problem setup (objective and constraints).
    - Key findings from the analysis.
    - Visualizations comparing original and adjusted scores.

General Guidelines

1. **Libraries Required:**
    - **CVXPY**: For solving optimization problems.
    - **Matplotlib/Seaborn**: For visualizing results.

2. **Report Structure (or power-point presentation):**
    - Brief introduction to the problem.
    - Description of the dataset and methodology.
    - Key insights and recommendations.
    - Visualizations with proper labeling and explanations.

3. ***In all projects,*** each team will **select a real dataset**, then **use CVXPY** to analyze the **constraints** and **make modifications** as described in the requirements.