Exercise 7: Financial Forecasting

Concept of Recursion

Recursion is a technique where a function calls itself in order to solve a problem. The problem is divided into smaller, simpler sub-problems, which are easier to solve. Each recursive call works on a smaller portion of the problem, eventually reaching a base case where the solution is straightforward.

Benefits of Recursion

Simplification: Recursion can simplify code and make it easier to understand, especially for problems that can be divided into similar sub-problems (e.g., tree traversals, factorial calculations).

Elegance: Recursive solutions can be more elegant and closer to the mathematical definition of a problem.

Recursive Structure

Base Case: The condition under which the recursion stops.

Recursive Case: The part where the function calls itself with a smaller or simpler version of the problem.

Time Complexity

Time Complexity: O(n), where n is the number of periods. This is because the algorithm makes a recursive call for each period.

Space Complexity: O(n) due to the call stack used for recursion. Each recursive call adds a new frame to the call stack.