# Q3: Comparison Analysis

## 1. Execution Time

• Q1: Vectorized (scikit-learn)  
 - Runs extremely fast.  
 - Uses optimized libraries (NumPy, BLAS, LAPACK).  
 - Execution time: typically < 0.01 seconds.  
  
• Q2: Non-Vectorized (Manual Gradient Descent)  
 - Much slower due to Python loops.  
 - No optimization from libraries.  
 - Execution time: can take 3–5 seconds or more for 1000 iterations.

## 2. Convergence Speed

• Q1: Vectorized  
 - Uses normal equation (one-step solution).  
 - Converges instantly to the optimal weights.  
  
• Q2: Non-Vectorized  
 - Needs many iterations to reduce error.  
 - Convergence depends on learning rate and number of iterations.  
 - May converge slowly or diverge if hyperparameters are poor.

## 3. Code Readability

• Q1: Vectorized  
 - Very clean and concise (5–10 lines).  
 - Easy to read, maintain, and scale.  
 - High-level API (e.g., LinearRegression()).  
  
• Q2: Non-Vectorized  
 - More verbose and manual.  
 - Better for educational purposes.  
 - Requires writing loops, updating gradients, and tracking cost.

## 4. Accuracy

• Q1: Vectorized  
 - Very accurate (exact solution for least squares).  
 - Typically results in a high R² score (~0.98–0.99).  
 - Stable and consistent results.  
  
• Q2: Non-Vectorized  
 - Accuracy depends on hyperparameters and initialization.  
 - May have slightly higher error if not tuned well.  
 - Can achieve similar accuracy with proper setup.

## 5. Summary

• Execution Time:  
 - Q1: Fast (<0.01s)  
 - Q2: Slower (few seconds)  
  
• Convergence:  
 - Q1: Instant  
 - Q2: Gradual (iterations needed)  
  
• Readability:  
 - Q1: High (simple and clear)  
 - Q2: Medium (manual logic)  
  
• Accuracy:  
 - Q1: High and stable  
 - Q2: Moderate to high (tunable)

# Q5: Conceptual Question

## 1. Multiple Linear Regression

• Definition:  
 - Predicts a single dependent variable using multiple independent variables.  
  
• Hypothesis Function:  
 - ŷ = θ₀ + θ₁x₁ + θ₂x₂ + … + θₙxₙ  
  
• Cost Function:  
 - J(θ) = (1/2m) \* Σ (ŷᶦ - yᶦ)²  
  
• Real-World Examples:  
 1. House Price Prediction  
 - Features: size, rooms, location score, year built  
 2. Student Grade Prediction  
 - Features: attendance, homework, midterm score, study hours

## 2. Multivariate Linear Regression

• Definition:  
 - Predicts multiple dependent variables from a shared set of independent variables.  
  
• Hypothesis Function:  
 - Ŷ = X · Θ  
  
• Cost Function:  
 - J(Θ) = (1/2m) \* Σ Σ (ŷⱼᶦ - yⱼᶦ)²  
  
• Real-World Examples:  
 1. Car Efficiency Prediction  
 - Predict city\_mpg and highway\_mpg using engine\_size, horsepower, etc.  
 2. Weather Forecasting  
 - Predict temperature, humidity, wind speed using time, lat, pressure

## 3. Key Differences Summary

• Outputs:  
 - Multiple Linear: One output  
 - Multivariate: Multiple outputs  
  
• Hypothesis Shape:  
 - Multiple Linear: Vector (m x 1)  
 - Multivariate: Matrix (m x k)  
  
• Model Usage:  
 - Multiple Linear: Predict single quantity  
 - Multivariate: Predict multiple related quantities  
  
• Example:  
 - Multiple Linear: Predict car price  
 - Multivariate: Predict city & highway mileage