**Summary of Findings and Reflections**

This report summarizes key findings and reflections from the Machine Learning Fundamentals II hands-on session focused on supervised learning, design matrices, cost functions, and gradient-based optimization.

We began by creating synthetic data that modeled a linear relationship between hours studied and exam scores. This served as an ideal dataset for understanding linear regression. A design matrix was constructed and normalized to ensure zero mean and unit variance, which facilitated faster convergence during optimization. Normalization was also applied to the real-world Boston Housing dataset for consistency in model performance.

Next, we calculated cost functions using Mean Squared Error (MSE) and Mean Absolute Error (MAE). MSE penalizes large errors more heavily than MAE due to the squaring of error terms. This makes MSE highly sensitive to outliers, whereas MAE is more robust. We tested different (w, b) values to observe how cost values changed, which highlighted the impact of parameter selection on model performance.

Gradient descent was then implemented from scratch to optimize a simple linear regression model. The algorithm minimized the MSE by iteratively updating the weights and bias. We observed that a learning rate of 0.01 provided stable convergence, whereas higher rates risked overshooting the minimum, and lower rates slowed down training. The learning curve showed a clear reduction in MSE over 100 iterations, validating the implementation.

Applying the same gradient descent method to the Boston Housing dataset yielded higher MSE and MAE compared to the synthetic data, largely due to the complexity and noise inherent in real-world data. This highlighted the limitations of simple linear models and emphasized the importance of preprocessing and feature selection.

Challenges encountered included selecting the appropriate learning rate and ensuring consistent scaling across datasets. We addressed this by experimenting with multiple rates and standardizing features using StandardScaler.

One potential improvement is to implement **mini-batch gradient descent**, which balances the speed of batch learning with the stability of stochastic methods. Additionally, comparing results with scikit-learn's built-in LinearRegression model provided a useful benchmark, demonstrating the efficiency of library implementations.

Overall, this session reinforced key ML fundamentals, including model design, cost analysis, optimization, and performance evaluation on both synthetic and real datasets.