**Regression Model Report**

The goal of this project is to predict house prices based on features such as square footage, number of bedrooms, bathrooms, and other property characteristics. We train two regression models and optimize their hyperparameters using efficient techniques.

**Models I Selected**

* **Gradient Boosting Regressor (GBR)**: A powerful ensemble method that builds trees sequentially, with each tree improving on the last. GBR is effective for capturing complex patterns in structured data.
* **K-Nearest Neighbors Regressor (KNN)**: An instance-based method that predicts values based on the average of nearby data points. KNN is simple but needs careful tuning to prevent overfitting.

**Hyperparameter Tuning: Bayesian Optimization**  
To optimize GBR, I used Bayesian Optimization via the Optuna library. This method intelligently explores the hyperparameter space, focusing on promising regions and reducing evaluations compared to traditional methods like Grid Search.

**Why I chose Bayesian Optimization**

* **Efficiency**: Reduces computational cost by narrowing the search to effective hyperparameter regions.
* **Informed Search**: Models the objective function to predict the best parameters.
* **Adaptability**: Refines the search based on previous results for faster convergence.

**Model Comparison**

* **GBR**: Excels at capturing complex relationships but can be computationally expensive and prone to overfitting without proper tuning.
* **KNN**: Simple and effective for capturing local patterns, but can struggle with high-dimensional data and requires careful tuning to avoid overfitting.

**Personal Experience**  
I encountered Bayesian Optimization in my Mathematics for AI course and was eager to apply it. It proved to be an excellent choice for hyperparameter tuning in this project.