Project Report - Phase 3: Model Development

Author: Arpit Jain **Date:** August 21, 2025 **Project:** Employee Productivity Prediction

3.1. Model Selection and Rationale

For this regression task, three different machine learning models were selected for training and comparison:

- **Linear Regression:** Chosen as a simple baseline model to establish a benchmark for performance.
- **Random Forest Regressor:** An ensemble learning method that is known for its high accuracy and ability to handle complex relationships in data.
- **XGBoost Regressor:** A highly optimized and powerful gradient boosting algorithm, which is often a top performer in machine learning competitions.

This selection provides a good range of models, from a simple baseline to more complex and powerful ensemble methods.

3.2. Training and Testing Methodology

The preprocessed dataset was split into a training set (80% of the data) and a testing set (20% of the data). This separation is crucial to ensure that the models are evaluated on data they have not seen before, providing an unbiased assessment of their performance. Each of the three models was trained on the same training data.

3.3. Model Performance Evaluation

The performance of each trained model was evaluated on the test set using the following standard regression metrics:

- **Mean Absolute Error (MAE):** This metric provides a straightforward measure of the average error of the model's predictions.
- **Mean Squared Error (MSE):** This metric penalizes larger errors more heavily.
- **R-squared (R^2) Score:** This is a key metric that indicates the proportion of the variance in the target variable that is predictable from the features. A higher R^2 score indicates a better fit.

3.4. Model Selection and Finalization

After comparing the evaluation metrics for all three models, the **XGBoost Regressor** was identified as the best-performing model, primarily due to its superior R^2 score. This indicates that it was the most accurate and reliable model for this particular dataset. The trained XGBoost model was then saved as a pickle file (best_model.pkl) so that it could be easily loaded and used in the web application without the need for retraining.