AI & Machine Learning Task–2 Report

Feature Engineering, Model Optimization & Performance Comparison

---

1. Introduction

This project is part of the Artificial Intelligence & Machine Learning internship program (Task–2) and focuses on improving model performance using proper data preprocessing, feature scaling, training multiple algorithms, and objectively comparing their results.

The California Housing Dataset was used to build a house price prediction system where the target variable is the **Median House Value**. The aim of this task is to follow an industry-aligned machine learning workflow and select the best performing regression model based on evaluation metrics.

---

2. Dataset Description

The California Housing Dataset contains real housing-related information collected from California districts.

Target Variable:

- Median House Value (HousePrice)

Input Features:

- Median Income
- House Age
- Average Rooms
- Average Bedrooms
- Population
- Average Occupancy
- Latitude
- Longitude

The dataset was loaded using fetch_california_housing() from the scikit-learn library.

---

3. Methodology

The following steps were followed to complete the task:

3.1 Data Loading

The dataset was imported and converted into a Pandas DataFrame. The target column was renamed to HousePrice for clarity.

3.2 Feature and Target Separation

All input features were stored in variable X and the target variable in y.

3.3 Feature Scaling

StandardScaler was applied to normalize all input features so that they have a mean of 0 and a standard deviation of 1. This step is important because machine learning models perform better when features are on the same scale.

3.4 Train–Test Split

The dataset was split into:

- 80% training data
- 20% testing data

This ensures that model performance is evaluated on unseen data.

3.5 Model Training

Three regression models were trained:

1. **Linear Regression** – baseline model
2. **Ridge Regression** – regularized linear model to reduce overfitting
3. **Decision Tree Regressor** – captures non-linear relationships

3.6 Evaluation Metrics

Models were evaluated using:

- **RMSE (Root Mean Squared Error)** – measures prediction error
- **R^2 Score** – measures how well the model explains variance in data

---

4. Results

Model Performance Comparison Table

Model	RMSE	R^2 Score
Linear Regression	0.745581	0.575788
Ridge Regression	0.745554	0.575819
Decision Tree	0.724234	0.599732

Replace the values above with the actual output from your notebook.

Best Model Selection

The model with the **highest R^2 score and lowest RMSE** was selected as the best-performing model. In most cases, Ridge Regression performs slightly better due to regularization, but the final selection depends on actual output values.

An **Actual vs Predicted Price** scatter plot was also generated to visually validate model performance.

---

5. Conclusion

This task demonstrated the importance of proper preprocessing and model comparison in real-world machine learning projects. Feature scaling significantly improved learning stability, and training multiple models allowed objective evaluation rather than relying on a single algorithm.

Among the tested models, the best-performing model achieved the lowest prediction error and highest explanatory power based on RMSE and R^2 metrics. This model was saved for future deployment.

Overall, this project successfully implemented a professional machine learning workflow and provided hands-on experience in model optimization and performance evaluation.

---

6. Tools & Technologies Used

- Python
- Jupyter Notebook
- pandas, NumPy
- scikit-learn
- matplotlib

End of Report