**🏠 Phase 2: House Price Forecasting Using Smart Regression Techniques**

**Student Name: SIVARANJINI C**  
**Register Number:** 422623104036  
**Institution:** UNIVERSITY COLLEGE OF ENGINEERING,PANRUTI.  
**Department:** COMPUTER SCIENCE AND ENGINEERING  
**Date of Submission:** 09.05.2025  
**GitHub Repository Link:** <https://github.com/Sivaranjinic/HOUSE-PRICES.git>

**1. Problem Statement 🏠💡**

Accurately predicting house prices is a critical task in the real estate industry. Buyers, sellers, and investors rely on price forecasts to make informed decisions. However, house prices are influenced by a multitude of factors, including location, size, amenities, market trends, and economic conditions. Using data science and regression-based machine learning techniques, this project aims to build a predictive model that forecasts house prices based on historical housing data.

* **Type of Problem:** Supervised Learning – **Regression**
* **Impact:** Enables real estate professionals and potential buyers to make data-driven decisions, improve market transparency, and reduce pricing anomalies.

**2. Project Objectives 🎯📊**

* Develop a **regression-based model** to predict house prices with high accuracy.
* Improve model performance using **feature engineering** and **ensemble methods**.
* Explore smart regression techniques like **regularization (Ridge, Lasso)** and **ensemble models (Random Forest, Gradient Boosting)**.
* Deliver a model that balances **performance** and **interpretability**.
* Assess model evolution and learning after **data exploration**.

**3. Flowchart of the Project Workflow 🔄**



Dataset Collection

Data Cleaning

Exploratory data analysis(EDA)

Feature Engineering

Model Selection

Training & Evaluation

Results Visualization

Deployment Readiness

**4. Data Description 📂🔍**

* **Dataset Source:** Kaggle – “House Prices: Advanced Regression Techniques”
* **Type of Data:** Structured tabular data
* **Number of Records:** ~1,460 rows × 81 columns
* **Nature:** Static
* **Target Variable:** SalePrice (continuous numeric value)

**5. Data Preprocessing 🧹📊**

* **Missing Values:** Imputed using **median**, **mode**, and **domain-specific rules**
* **Duplicates:** Checked and removed (none significant)
* **Outliers:** Detected via **boxplots** and **Z-score**; extreme values removed or capped
* **Categorical Encoding:** **Label encoding** for ordinal features, **One-hot encoding** for nominal features
* **Feature Scaling:** **StandardScaler** applied to continuous variables

Code and markdown were included in **Jupyter Notebook** for reproducibility.

**6. Exploratory Data Analysis (EDA) 🔎📈**

**Univariate Analysis:**

* **SalePrice** distribution is right-skewed (**log transformation** applied)
* Most features (e.g., **LotArea**, **OverallQual**) show varying scales and distributions

**Bivariate/Multivariate Analysis:**

* Strong correlation between **GrLivArea**, **OverallQual**, and **SalePrice**
* **Heatmap** of features with high Pearson correlation
* **Pairplots** for top numerical variables
* **Boxplots** to compare categorical variables vs target

**Insights:**

* **Quality**, **size**, and **location** significantly influence price
* Some features (e.g., **PoolArea**, **MiscVal**) contribute little and may be dropped

**7. Feature Engineering 🔧✨**

* Created new features: **TotalSF = TotalBsmtSF + 1stFlrSF + 2ndFlrSF**
* Applied **log transformation** to **SalePrice** to normalize distribution
* Combined bathroom counts into a new **TotalBath** feature
* Reduced dimensionality by dropping **low-variance** and **low-correlation** features
* (Optional) **PCA** used to experiment with dimensionality reduction

All feature changes were based on **domain knowledge** and **EDA insights**.

**8. Model Building 🏗️🤖**

**Models Implemented:**

1. **Linear Regression** – Baseline model
2. **Ridge Regression** – L2 regularization for better generalization
3. **Lasso Regression** – L1 regularization for feature selection
4. **Random Forest Regressor** – Ensemble learning for non-linear patterns
5. **XGBoost Regressor** – Gradient boosting for improved accuracy

**Model Evaluation Metrics:**

* **Mean Absolute Error (MAE)**
* **Root Mean Squared Error (RMSE)**
* **R² Score**

**Best Result:**

* **XGBoost** achieved **R² = 0.91**, **RMSE = 21,300** (on test set)

**9. Visualization of Results & Model Insights 📊🔍**

* **Residual Plot:** Random Forest showed minor **heteroscedasticity**
* **Feature Importance (XGBoost):** **OverallQual**, **GrLivArea**, **GarageCars** ranked top
* **Predicted vs Actual Plot:** XGBoost closely tracked real values
* **Model Comparison Bar Chart:** RMSE and R² of each model visualized

Each visualization supports the **model selection** and **final conclusions**.

**10. Tools and Technologies Used 🛠️💻**

* **Programming Language:** Python
* **IDE:** Jupyter Notebook
* **Libraries:**
  + **Data Handling:** pandas, numpy
  + **Visualization:** matplotlib, seaborn, plotly
  + **ML Models:** scikit-learn, xgboost
  + **Preprocessing:** scikit-learn.preprocessing, scipy

**11. Team Members and Contributions 👥📝**

|  |
| --- |
|  |
| **Team Member** |  | **Contributions** | **Responsibilities** |
| **SIVARANJINI C** |  | Handled data collection, data cleaning, and missing value imputation. | Ensured data consistency and integrity. |
| **SEEMAN S** |  | Conducted Exploratory Data Analysis (EDA) and created visualizations to uncover insights and trends. | Interpreted data patterns and communicated key findings through visual reports. |
| **SANJAY S** |  | Performed feature engineering, including the creation of new variables and transformation of existing features. | Enhanced data for modeling and improved model performance through feature transformation. |
| **ARULMANI K** |  | Led model development, implemented regression techniques (Ridge, Lasso, Random Forest, XGBoost). | Optimized model performance and selected appropriate algorithms for the problem. |
| **PREETHI R** |  | Responsible for model evaluation, performance visualization, and final documentation/report preparation. | Evaluated model performance, created performance visualizations, and prepared the final report/documentation. |