**House Prices - Advanced Regression Techniques**

Group 2: Faisal Rahman Chowdhury

Dec 01, 2024

**Abstract**

The project is focused on predicting house prices utilizing advanced machine learning techniques. This is Kaggle Competition challenge where a dataset with 81 features was given to apply different regression models such as Linear Regression, Random Forest, Gradient Boosting, and XGBoost. I have utilized my knowledge of feature engineering, data cleaning, data preprocessing, training regression models and performance evaluation with MAE, MSE, EMSE and R-Square metrics. The final results shows that Random Forest is the best model because of its capacity to manage complex relationship having higher R-square and less errors.

1. **Introduction**

Machine learning is a subject of Artificial Intelligence (AI). In this project, I have used machine learning models to predict housing prices. Price of houses are determined by various factors such as material quality, size of living area, size of garage, number of bedrooms, parking size and so on. I will be using multiple regression models and identify the most optimal model with maximum R-Square value to predict the house prices.

**1.1 The problem I tried to solve**

Real estate is one of the key sectors of national economy and also a major concern for citizen. With the increasing demand of housing, prices are also soaring in united states. Often it is critical to provide accurate predictions of housing prices. If it is possible to make accurate predications in this sector it could lead to huge developments in real estate pricing, taxation and market trends. I will try to solve the problem using supervised machine learning.

**1.2 Related Work**

The project was based on the Kaggle competition shared dataset. A massive amount of ideas and methodologies have already been developed for the same problem in a similar domain by other participants.

Ensemble methods like Random Forest and Gradient Boosting, in particular, played a prominent role across successful submissions. These approaches have been well-known for their stability in regression problems, particularly when several features have a complex association.

**1.3 Available tools and programs**

The following tools and libraries were utilized:

* **Scikit-learn**: Building and evaluating regression models
* **Pandas and NumPy**: Data cleaning and Data preprocessing for model development.
* **Matplotlib and Seaborn**: Data Visualization and interpretations.
* **XGBoost**: Robust gradient boosting models.  
    
  These tools and techniques were highly efficient for this particular project.

1. **Overview of the architecture**

This project is using a modular architecture that includes a series of connected modules for data preprocessing, feature selection, machine learning (ML) model training, and performance evaluation. These modules collaborate to deliver an efficient workflow for predictive modeling*.*

**2.1 Finished work: Running modules**

 Data Preprocessing:

 Feature Selection:

 Model Training:

 Final **Model Selection:**

 Evaluation Metrics:

 Prediction and Submission:

1. **Data Collection**

The dataset was sourced from Kaggle’s competition on **House Prices - Advanced Regression Techniques**. It included 1460 rows and 81 features describing various property attributes. Data preprocessing addressed categorical variables, and features engineering.

1. **Baseline and Proposed Methods**

**Baseline Method**

**Linear Regression** used as the baseline model to compare the performance. It is a very interpretable, yet simple model, that did not perform well by capturing the non-linear relationships inherent in the dataset.

**Proposed Methods**

* **Random Forest Regressor:** Selected for its ability to handle nonlinear relationships and its robustness to overfitting. This model fits well, capturing complex relationships in the data.
* **Gradient Boosting:** Used to build a predictive model in the form of an ensemble that combines weak predictive models in a relevant manner, usually sequentially where subsequent models try to correct the errors of previous models.
* **XGBoost:** An extension of Gradient Boosting that has become very popular due to its high performance and speed, this algorithm is great in handling larger datasets.
* **Regularized Models:**
  + - **Ridge Regression:** Applied to reduce overfitting by adding an L2 penalty, which shrinks the coefficients of less important features, while keeping all features in the model.
    - **Lasso Regression:** Used for feature selection by adding an L1 penalty, which encourages sparsity and reduces the impact of irrelevant features.
    - **ElasticNet Regression:** A hybrid model combining both L1 and L2 penalties, offering a balance between feature selection (Lasso) and coefficient shrinkage (Ridge).

1. **Implementation**

Data Preprocessing:

* Used pandas’ get\_dummies function to handle categorical variables by converting them into dummy/indicator variables.
* Managed missing values and ensured the dataset was clean for training and evaluation.

 Feature Selection:

* Conducted correlation analysis to identify variables that strongly influence the target (SalePrice). Correlation helped in understanding relationships between independent variables and the target.
* Set a cross-correlation threshold of 0.5 for filtering strong features This guaranteed that the dataset was concentrated on predictors that provided the most explanatory power, while all less relevant variables were filtered out.

A screen shot of a computer code

Description automatically generated

* Identify highly correlated features from the relevant features for the target value.

A screenshot of a graph

Description automatically generated

* Removed highly Correlated features (Set threshold 0.8) to avoid multicollinearity. It ensures that model did not face redundancy or instability in predictions due to overlapping information.

A screenshot of a computer program

Description automatically generated

* Used Random Forest Feature Importance to further refine the selection of predictors (X values). The importance scores guided the elimination of less relevant features, improving the model’s efficiency and performance.

A screenshot of a computer program

Description automatically generated  
A close-up of a message

Description automatically generated

 Model Training:

 Explored multiple models to identify the most effective approach for predicting house prices:

* **Linear Regression:** Implemented as a baseline model but found to perform inadequately due to the non-linear nature of the dataset.
* **Lasso Regression:** Applied to add regularization and reduce the complexity of the model. However, it did not perform as well as other ensemble methods due to its tendency to eliminate important features.
* **ElasticNet Regression:** Combined the strengths of both Lasso and Ridge regression by penalizing large coefficients. It showed reasonable performance but still fell short compared to the ensemble methods.
* **Random Forest Regressor:** Used for its ability to handle non-linear relationships and robustness against overfitting. It was the top performer in terms of accuracy and stability.
* **Gradient Boosting:** Evaluated for its effectiveness in improving prediction accuracy through sequential learning. However, Random Forest performed similarly with fewer computational costs.
* **XGBoost:** Applied as a powerful gradient boosting technique, but Random Forest achieved similar performance with simpler tuning.

I have also used Grid Search for Regularized Models (Ridge, Lasso and Elastic Net) and boosting models (Gradient Boosting and XGBoosting) to get better performance by hyper tuning. The purpose of using best parameter to not miss any important parameter in terms of building a good model that can outperform.

A screenshot of a computer program

Description automatically generated

 Final **Model Selection:**

* Chose **Random Forest Regressor** as the final model due to its balance of performance, interpretability, and computational efficiency.

 Evaluation Metrics:

* Utilized Mean Absolute Error (MAE), Mean Squared Error (MSE), R2 Score, and Root Mean Squared Error (RMSE) to assess the model's performance on the test data.

 Prediction and Submission:

* Predicted housing prices for the dataset and created a submission-ready file by including the Id and corresponding predicted SalePrice.

1. **Results and Evaluation**

The running code trains multiple models, including **Random Forest**, **Gradient Boosting**, **XGBoost**, and regularized linear models (**Ridge**, **Lasso**, **ElasticNet**), on a housing price prediction task. The models were evaluated using **Root Mean Squared Error (RMSE)**, **Mean Squared Error (MSE)**, and **R-squared (R²)** metrics, which were calculated on the logarithm of the actual and predicted sale prices, as required for the Kaggle competition.

**Model Performance:**

* **Random Forest Regressor:** This model performed exceptionally well, capturing the non-linear relationships in the dataset. Hyperparameter tuning, including adjustments to n\_estimators and max\_depth, helped improve performance.
  + **RMSE:** 30115
  + **MAE:** 19212
  + **R²:** 0.88  
      
    The model demonstrated robustness, with a reasonable RMSE on the test set and a high R², indicating that a significant portion of the variance in the target variable was explained.
* **Gradient Boosting:** Gradient Boosting was slightly more prone to overfitting on the training data, but with fine-tuning of the learning rate, it provided strong results. It was less robust than Random Forest, especially on the more varied test set.
  + **RMSE:** 30,355
  + **MAE:** 19207
  + **R²:** 0.87  
    The model was a close competitor to Random Forest but showed marginally worse performance on the validation set.
* **XGBoost:** XGBoost, using early stopping, was highly efficient and performed well in preventing overfitting. It delivered competitive results with the Random Forest model, often performing slightly better due to its ability to fine-tune during training. However, it was computationally more intensive than the other models.
  + **RMSE:** 30,160
  + **MAE:** 19,299
  + **R²:** 0.88  
      
    XGBoost slightly outperformed both Random Forest and Gradient Boosting, showing superior predictive performance with a lower RMSE and MSE.
* **Regularized Linear Models (Ridge, Lasso, ElasticNet):** These models, though effective in penalizing large coefficients and preventing overfitting, did not perform as well as the tree-based models. They struggled to capture complex relationships between features. The Lasso model showed some potential for feature selection, but the overall performance was lower compared to the more powerful ensemble methods.

**Advantages and Disadvantages:**

* **Advantages:**
  + The **ensemble methods** (Random Forest, Gradient Boosting, XGBoost) showed clear advantages in capturing non-linear relationships and were more robust to overfitting compared to linear models.
  + **XGBoost** and **Gradient Boosting** provided high performance by focusing on sequential error correction, whereas **Random Forest** was effective in preventing overfitting without the need for fine-tuning as much.
  + **Regularization** techniques like **Ridge** and **Lasso** helped reduce the risk of overfitting, but their performance was limited due to the underlying linear assumptions, which couldn't capture the complexities of the data as effectively as tree-based models.
* **Disadvantages:**
  + **Tree-based models** (Random Forest, Gradient Boosting, XGBoost) tend to be computationally expensive and may require more time and resources to train, especially on large datasets.
  + **Linear models** (Ridge, Lasso, ElasticNet) were less effective in this case because the dataset exhibited complex relationships between features that linear models could not adequately capture.
  + **Gradient Boosting** was prone to overfitting on noisy data, requiring careful hyperparameter tuning, while **XGBoost** added computational complexity due to early stopping mechanisms and regularization.

**Comparison with Benchmark:**

The models performed better than the baseline **Linear Regression** model, which was simple but struggled with non-linear relationships in the dataset. Linear Regression Metrix:

* + **RMSE:** 38,749
  + **MAE:** 24,074
  + **R²:** 0.80

Compared to other groups in this course, the ensemble methods provided a more generalized and higher-performing solution. However, due to the computational cost of tuning models like XGBoost, performance could be slightly slower compared to other simpler models

1. **Achievements and Observations:**

**Contributions**:

* 1. Data preprocessing pipeline development.
  2. Implement Linear Regression
  3. Implementation of Random Forest and Gradient Boosting.
  4. Conducted Regularized Linear Models
  5. Evaluation and submission of predictions to Kaggle.

A screenshot of a computer

Description automatically generated

**A screenshot of a phone

Description automatically generated**

**Learnings**:

* 1. Gained insights into ensemble methods for regression.
  2. Improved understanding of preprocessing techniques, feature engineering and hyperparameter tuning.
  3. Building model and feature engineering on training data set, let use the model and technique on the test dataset to predict values.

1. **Discussion and Conclusions**

The project successfully predicted house prices using advanced machine learning techniques. Random Forest emerged as the most effective model, achieving a good balance between performance and interpretability. Future improvements could include using neural networks, additional data sources, and advanced optimization techniques to enhance model accuracy further.