Capstone project report

APARTMENT PRICES PREDICTION

Machine Learning - IT3190E

Group 5 - DSAI

Nguyen Hai Long 20214911

Ha Hoang Hiep 20214891

Vu Duc Hung 20214902

Hanoi University of Science and Technology

School of Information and Communication Technology

Table of contents

I. Introduction.

Real estate valuation is an important activity for buyers and sellers who want to estimate the value of houses and lands. In today's modern society, people have higher living standards and demand more than just a place to live. Owning houses can provide financial benefits and improve the quality of life. Therefore, we aim to build a machine learning system that can predict the value of apartments in Hoang Mai district (Hanoi), utilizing scikit-learn library's regression models.

II. Features selection.

The price of an apartment, in fact, depends on many factors like area, number of floors, number of rooms, the position (near school, market, avenue, etc).

Initially, the crawled data set was composed of 5 features: the area, the number of bedrooms, the floor which the apartment is on, the width of the road and whether it has a parking place or not.

But after visualizing the data, we decided to choose only two features: the area and the number of bedrooms. The decision to choose only these two features was based on the analysis of the correlation matrix heatmap and the boxplot of data points for all the features as below.

Figure 2. The heatmap of visualized data.

By examining the correlation matrix heatmap, we can identify the relationships between each feature and the target variable ('price'). The heatmap displays the correlation coefficients, which indicate the strength and direction of the linear relationship between variables.

The heat map showed that 'area' and 'bedroom' have the highest positive correlations with 'price'. This indicates that as the values of 'area' and 'bedroom' increase, the 'price' tends to increase as well. Other features displayed weaker correlations or no significant relationship with 'price'.

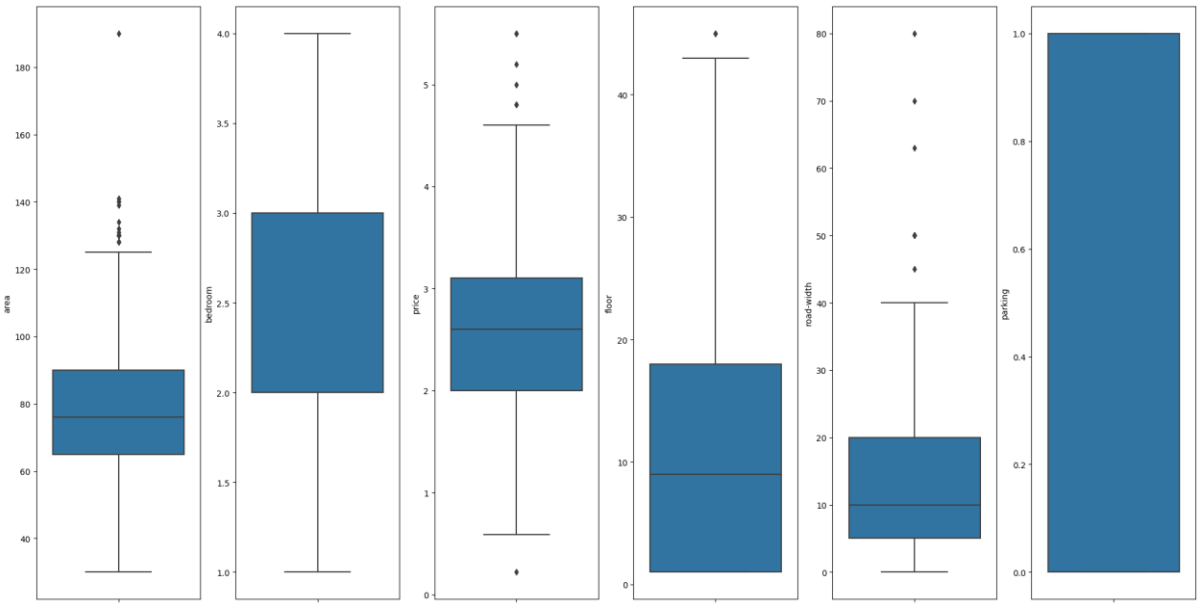


Figure 2. The box plot of data points.

The boxplot provides a visual representation of the distribution of data points for each feature. It helps in understanding the range, central tendency, and presence of outliers in the data. The boxplot analysis revealed that 'area' and 'bedroom' exhibit the most noticeable variations in their datapoint distributions. Other features either had limited variability or did not exhibit clear patterns.

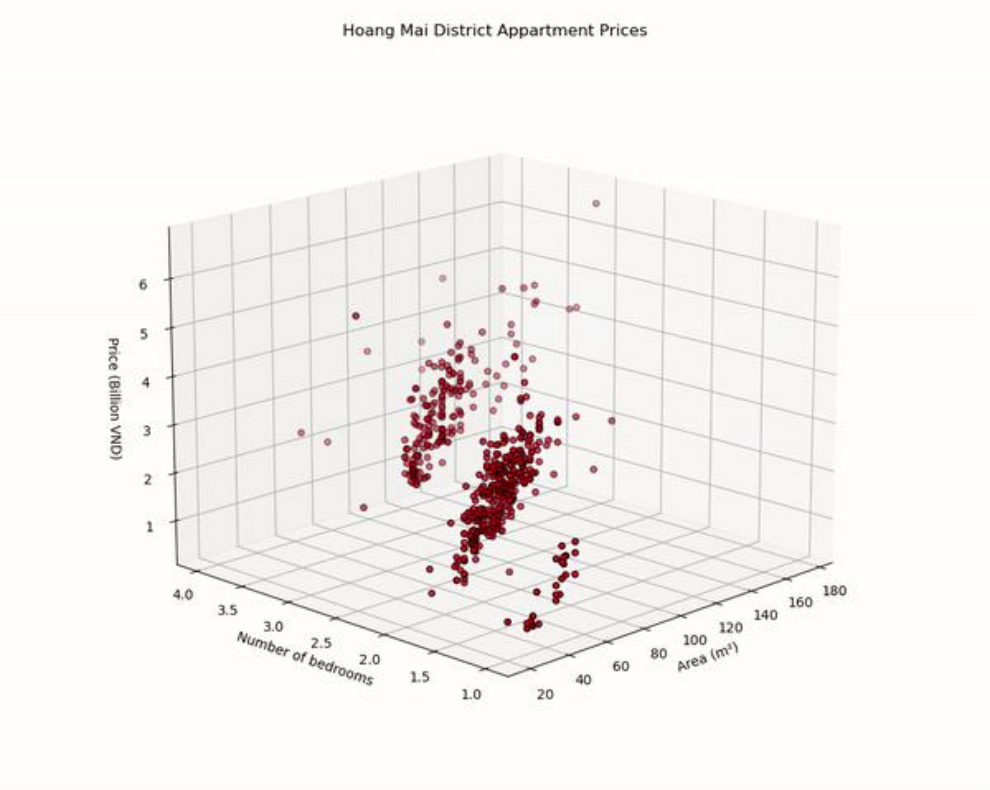


Figure 3. Visualized data with two features (area and bedroom).

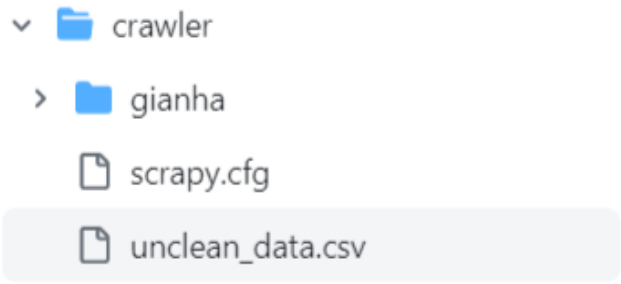
Considering these observations, we selected 'area' and 'bedroom' as the features for our analysis. These features showed the strongest correlations with 'price' and displayed significant variations in their datapoint distributions. This feature selection process aimed to focus on the most influential variables and simplify the model while capturing the key factors affecting the 'price' prediction.

III. Implements and dataset.

The main file of this project is the dataset.csv, which contains crawled data for the project, and the regression_models.py, which is the source code of our project (including all the models we will use). About the other files and folders:

- *select_best_forest*: the method to find the best number of trees for the Random forest regression.
- *select_best_k*: the method to find the best k for the K-Neighbor regression.
- *crawler*: contains the uncleaned data and scrapy crawling method.
- *datacleaner.py*: is the data cleaning method.
- *features_selection*: explain why we choose our two features.

Our data set was taken at https://alonhadat.com.vn/ on April 19th, 2023 through our own web crawling efforts (using Scrapy). The raw (unclean) data was saved in the *unclean_data.csv* file.



The crawled data will be in the *dataset.csv* file, which will be used for both training set and testing set. The system will randomly choose 80% of the data for training and the rest for testing.

IV. Methodology.

In this project we will use regression methods to analyze the price of an apartment: Linear, Ridge, Lasso regression; K-Neighbor regression and Random forest regression. The models' performance was evaluated using Mean Squared Error (MSE), defined below as:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2.$$

where:

- n: the sample of data points on all variables.
- Y: the observed values of the variable being predicted
- $\hat{Y}$: the predicted values

Let's go into each model.

1. Linear, Ridge and Lasso regression.

   Given a training set D = $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2)\}$, in which:

   - y = f(x): perform the price of an apartment.
   - $\mathbf{x}_1$, $\mathbf{x}_2$: the vectors represent the area and the number of bedrooms, respectively.

   **Linear regression:* we need to learn a function y = f(x)

   The regression model:

   $$f(x) = w_0 + w_1x_1 + w_2x_2$$

   **Ridge regression:* with given set D we need to solve:

$$f^* = \arg\min_{f \in \boldsymbol{H}} RSS(f) + \lambda \|\boldsymbol{w}\|_2^2$$

$$\Leftrightarrow \boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \sum_{i=1}^{M} (y_i - \boldsymbol{A_i w})^2 + \lambda \sum_{j=0}^{n} w_j^2$$

where:

- Ai = (1, x_{i1}, x_{i2}, …, x_{in}) is composed from xi
- λ is a regularization constant (λ > 0).
- $\|w\|_2$ is the L^2 norm.

then the prediction for a new **x**: $y_x = w_0^* + w_1^* x_1 + w_2^* x_2$

**Lasso regression:* By replacing L^2 norm from Ridge regression by L^1 norm, we will get the Lasso regression:

$$w^* = \arg\min_{\boldsymbol{w}} \sum_{i=1}^{M} (y_i - \boldsymbol{A_i w})^2 + \lambda \|\boldsymbol{w}\|_1$$

2. <u>K-Neighbor regression.</u>

Algorithm:

To predict the price of a new apartment z:

+ For each element x in the training set D, the KNN model will calculate the distance between x and z.
+ Determine the set *NB*(z) (which is the k nearest neighbors of z calculated by a distance function d)
+ Predict the output value of z: $y_z = \frac{1}{k} \sum_{\boldsymbol{x} \in NB(\boldsymbol{z})} y_x$

Determining the optimal k:

```
# Iterate over the k values
for k in k_values:
    # Initialize weighted and unweighted KNN regressors
    weighted_knn = KNeighborsRegressor(n_neighbors=k, weights='distance')
    unweighted_knn = KNeighborsRegressor(n_neighbors=k)

    # Fit the models
    weighted_knn.fit(x_train, y_train)
    unweighted_knn.fit(x_train, y_train)

    # Predict on the test set
    weighted_knn_preds = weighted_knn.predict(x_test)
    unweighted_knn_preds = unweighted_knn.predict(x_test)

    # Calculate mean squared error for weighted and unweighted KNN
    weighted_knn_mse[k] = mean_squared_error(y_test, weighted_knn_preds)
    unweighted_knn_mse[k] = mean_squared_error(y_test, unweighted_knn_preds)
```

In order to determine the optimal k value, we divided the data set into a training set and a validation set. And after trying k from 1 to 20, we get the optimal k = 15.

3. Random forest regression.

**Algorithm*: with input is learning set D, number of trees k

+ Create k trees, each tree is generated by building a subset Di by randomly taking (with duplicates) from D.
+ Learn the i-th tree from Di as follows: At each vertex (node), randomly select a subset of attributes and branch the tree based on that attribute set (The tree will be born to its full size, without pruning).
+ Each subsequent judgment is obtained by averaging the statements guessed from all the trees.

**Optimal parameters*: We divided the data set into a training set and a validation test. After gridsearch, we found the optimal parameters

- Number of trees in the forest is k = 100.
- Maximum depth of the trees is 5.
- Minimum number of samples required to split a node is 10.
- Minimum number of samples required at each leaf node is 2.

V. Results and analysis.

Method	MSE (negative)
Linear Regression	- 0.271796
Ridge	- 0.271798
Lasso	- 0.276118
Weighted K-Neighbors	- 0.269898
K-Neighbors	- 0.262718
Random Forest	- 0.251574

Table 1. Mean Square Error (MSE) of different methods (higher is better).

In the analysis conducted, several regression methods were compared to predict house prices. Among them, Ridge regression and linear regression

exhibited similar performance (MSE is about - 0.271). This similarity can be attributed to the dataset's characteristics, which may not include strong multicollinearity or overfitting issues. Since strong multicollinearity was not present, the regularization term in Ridge regression had minimal impact, resulting in comparable performance between the two methods. Consequently, Ridge regression, which includes regularization, and linear regression produced similar results, suggesting that regularization does not provide significant benefits in this scenario.

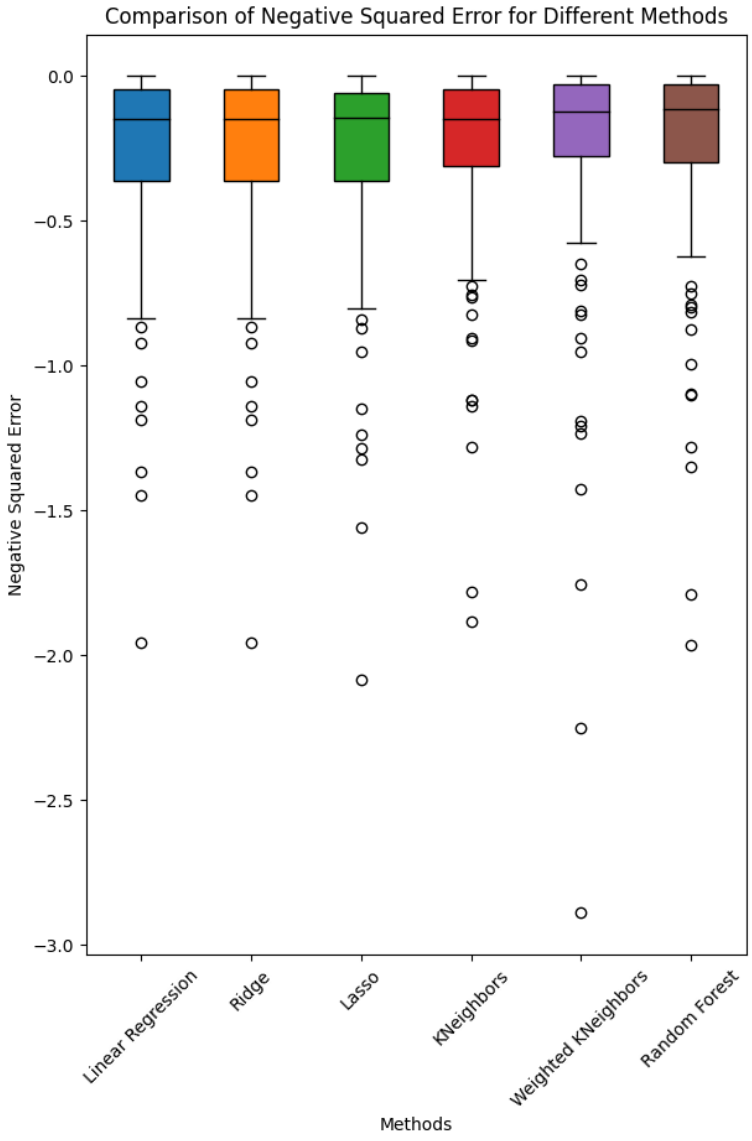


Figure 5. Compare the Negative Square Error for different methods

On the other hand, Lasso regression showed inferior performance compared to Ridge regression and linear regression (MSE = -0.276). Lasso regression uses L^1 regularization, which encourages sparsity by

driving some coefficients to exactly zero. However, in this analysis, a feature selection process was conducted prior to the regression analysis. This process aimed to identify and include the most relevant features, resulting in a reduced number of irrelevant or redundant features. As a result, the dataset had already undergone a form of feature selection, making Lasso regularization less beneficial. Therefore, Ridge regression and linear regression outperformed Lasso in this scenario.

The underperformance of linear models, including Linear Regression, Ridge Regression, and Lasso Regression, can be attributed to the dataset's characteristics. It is possible that the relationship between the selected features and house prices exhibits non-linear patterns that cannot be effectively captured by linear models. Linear models assume a linear relationship between the features and the target variable, which may limit their ability to accurately predict house prices when non-linear relationships are present in the data.

We chose not to optimize the alpha parameter for Ridge and Lasso regression (using a fixed $\alpha = 0.5$) because linear regression without regularization consistently performed better. Increasing alpha worsened the results, while decreasing it to near zero was pointless. Hence, we prioritized the simplicity and interpretability of non-regularized linear regression.

KNN model's performance is evaluated using the k nearest neighbors' values. KNN is sensitive to the choice of k and the distance metric. In this scenario, the chosen configuration of KNN regressors, with the weighted distance and a suitable pre-determined $k = 15$, can be considered more optimal for predicting house prices, resulting in lower MSE (- 0.263 ~ - 0.269) compared to Linear, Ridge and Lasso Regression.

In contrast, Random Forest Regression emerged as the most superior method for predicting house prices (MSE = - 0.251). The ensemble nature of Random Forest helps capture complex relationships and interactions between features, resulting in robust predictions. Additionally, Random Forest Regression excels in handling both linear and non-linear relationships between features and the target variable. After performing a

grid search to optimize the model, we found that adjusting the following key parameters further improved its predictive performance: the number of trees in the forest, the maximum depth of the trees, the minimum number of samples required to split a node, and the minimum number of samples required at each leaf node. Specifically, the optimal parameter values we obtained were: Number of trees in the forest is 100, maximum depth of the trees is 5, minimum number of samples required to split a node is 10, and minimum number of samples required at each leaf node is 2. These parameter adjustments ensure a well-balanced trade-off between model complexity and generalization ability, leading to enhanced accuracy and reliable predictions.

VI. Difficulties and proposed solutions

1. Difficulties:

- *Data realism*: Real estate data obtained from web crawling is often not large and may not exhibit a smooth pattern suitable for regression analysis. This is because it reflects real-world properties with variations due to factors (such as market dynamics, …)
- *Inaccurate data:* Data from https://alonhadat.com.vn/ sometimes contains typing errors. For example: the width of the road is 80 meters,..Inconsistent data format:
- *Inconsistent data formats*: particularly regarding prices. For example, prices may be expressed in different units such as "tỷ" (billion), "triệu" (million), or "triệu/m2" (million per square meter).
- *Missing data:* Real estate data often contains missing values. This can be a problem for regression models, as they require complete data to learn.

54	68 m	2 phòng ngủ	1,75 tỷ
55	92 m	3 phòng ngủ	3,6 tỷ
56	55 m		1,6 tỷ
57	66 m	2 phòng ngủ	1,77 tỷ

Figure 6. Example of missing data

- *Outliers:* Real estate data often contains outliers. These are data points that are far outside the normal range (for example: the price is too large, etc). Outliers can skew the results of regression models, so we have to remove them before training our model.

2. Proposed solutions:

- Feature engineering: Feature engineering can transform data in a way that makes the relationships between the features and the target variable more suitable. In this project we have selected two features that mostly affect the price of an apartment to make the data more linear.
- For missing data, we just simply remove the data containing missing values.
- Outlier detection: By using outlier detection techniques to identify and remove outliers, we have removed all the apartments having more than 5 bedrooms and having the price over 20 billions from our data set.

References:

- *sklearn*
- *https://alonhadat.com.vn/*
- *The Boston Housing Dataset | Kaggle by Prasad Perera*