Contents

Abstract: 3
1. Introduction. 4
2. Literature Review. 4
3. Data Science Process 5
3.1 Problem statement. 5
3.2 Dataset collection 6
5. Features of the dataset. 6
6. Machine learning algorithm. 6
6.1 Random Forest (RF). 6
6.2. Support Vector Regression (SVR) Machine learning Algorithm. 8
6.3. XGBOOST Machine Learning Algorithm 8
7. Dataset Importation. 10
8. Data Cleaning. 10
9. Exploratory data analysis (EDA). 11
1O. Feature Engineering Analysis 12
10.1. One hot encode. 12
10.2. Row reduction. 12
11. Modeling: 13
11.1. Dataset split: 13
11.2. Model Training: 13
11.3. Validation: 13
11.4. Cross - validation: 13
11.5. Testing 13
12. Discussion: 17
13. Conclusion. 18
14. Reflection on professional, ethical, and legal issues in machine learning and house price dataset. 18
References: 19

Development of house price prediction models using machine learning algorithms

Eze, Jonas Chinweike(Q2156726)

Abstract:

House price refers to the monetary amounts at which houses can be purchased or sold. This study developed house price prediction machine learning models using three algorithms: Support Vector Regression, Random Forest and XGBoost. During the development, the dataset was split into three sets: 70% for training,15% for validation and 15% for testing. The hyperparameters of the algorithms were tuned using grid search method. Each of them had three configurations, the models were evaluated using Mean Square error (MSE), Mean Absolute Error (MAE) and Coefficient of Determination(R^2) and the hyperparameters with the best performance configuration was printed. Also, they were cross-validated using k-fold validation method. From the performance evaluation metrics results, XGBoost performed better with Coefficient of Determination (R^2) value of 0.601, MAE value of 37392.80 and MSE value of 2211439098.71 on the test set while Random Forest followed it with Coefficient of Determination (R^2) value of 0.593, MAE value of 37775.51 and MSE value of 2255574714.14 on the test set. Support Vector Regression achieved the least result with R^2 value of 0.04031, MAE value of 60284.8831 and MSE value of 5324250823.44 on the test set. While there are limitations to be considered, XGBoost and Random Forest findings provided valuable insights for investors, realtors, banks, and government agencies. Also, the study provides avenues for future researchers.

1. Introduction.

According to Collins English Dictionary, the term "house price" refers to the monetary sums involved in the buying or selling of houses. In many countries, the determination of house prices relies on indices such as the US Federal Housing Finance Agency HPI, UK National Statistics HPI, UK Land Registry's HPI, UK Halifax HPI, and UK Rightmove HPI. These indices, like the HPI, utilize weighted, repeat sales data to gauge average price changes in properties over time. The data for these indices are sourced from mortgage transactions on single-family properties backed by Fannie Mae or Freddie Mac since January 1975. While the HPI offers insights into mortgage defaults, prepayments, and housing affordability in specific regions, it's not suitable for predicting the price of individual houses due to its broad nature. Other factors like location, age, and structural features must also be considered. (Truong et al.,2020). In some countries, traditional methods of predicting house prices rely on physical conditions, location, and other features like the number of bedrooms. However, these methods may be influenced by agents seeking their own benefit, such as through agent fees. To overcome these limitations and effectively predict house prices considering various features, a machine learning approach is necessary. (Ahtesham et al.,2020).

Machine learning is a sub field of Artificial Intelligence (AI) that focus on the development of an algorithm and a statistical model that learn from data without being explicitly programed. So, the algorithm is trained using data(features) and it makes prediction(decision) using the model(knowledge) gained during the training.

2. Literature Review.

The scientific community is currently involved in developing methods and tools to identify and predict different housing prices, which have significant implications for people's welfare. Several researchers have utilized machine learning methods to forecast house prices, as exemplified by:

To efficiently predict house price, Park and Bae (2015) applied machine learning algorithms as a research methodology to develop a housing price prediction model. In their study, different machine learning algorithm were used. These include C4.5, RIPPER, Naïve Bayesian, and AdaBoost. The result in all the tests showed that RIPPER outperformed the other housing price prediction models.

Also, Ho et al. (2020) in their work to develop a model for property price prediction in Hong Kong using machine learning algorithms employed Support Vector Machine (SVM), Random Forest (RF) and Gradient Boosting Machine (GBM). These algorithms were examined and compared using three performance metrics: Mean Squared Error (MSE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE). Following the examination results, it was observed that RF and GBM achieved better performance when compared to SVM.

Another analysis by Zulkifley et al. (2020) developed machine learning house price prediction models using five different algorithms which include Support Vector Regression (SVR), Artificial Neural Network (ANN), XGBoost and Multiple Linear Regression (MLR). The models were examined using Root Mean Squire Error (RMSE) metrics. First, they were evaluated with locational and structural attributes and secondly, they

were examined with locational attribute only. The result showed that the models achieved better results with locational attribute only and SVR out performed other models with lowest RMSE value of 0.0047 followed by ANN with RMSE value of 0.058.

In addition, Ravikumar (2018) in his research work developed Real Estate Price Prediction models using Machine learning algorithms like a Random Forest (RF), Multiple Regression (MR), Support Vector Machine (SVM), Gradient Boosted Trees (GBT), Neural Networks, and Bagging. The models were evaluated using two performance metrics: Root Mean Squire Error (RMSE) and Accuracy. Following the results, RF showed a better performance with accuracy value of 90% and RMSE value of 0.012 followed by Bagging which achieved accuracy value of 70% and RMSE value of 0.563. However, Multiple Regression showed least performance with accuracy value of 55% and RMSE value of 0.70.

Again, Jha et al. (2022) in their research work to develop efficient and effective house price prediction model applied nine different machine learning algorithms such as VCPA Model, Linear Regression, support vector regression (SVR), Decision Tree, Random Forest, XGBoost , Lasso, Voting Regressor and CatBoost. The models were evaluated using three difference performance matrices: Coefficient of Determination (R^2), Mean Square Error (MSE), Mean Absolute Error (MAE). Result showed that XGBoost Performed better with R^2 value of 0.92, MSE value of 8.59 and MAE value of 4.1, followed by Random Forest with R^2 value of 0.91, MSE value of 8.79 and MAE value of 4.2 while VCPA Model had least performance with R^2 value of 0.74, MSE value of 9.36 and MAE value of 4.57.

3. Data Science Process

Problem identification ⇒ Raw dataset collection ⇒ Clean/pre-processing of raw dataset ⇒ Exploratory data analysis ⇒ Build model and Analyze result ⇒ Presentation of result in a visual way.

3.1 Problem statement.

According to Investopedia (2024), fluctuations in house prices can significantly impact the economy. Rising house prices typically stimulate job creation, bolster confidence, and encourage higher consumer spending, leading to increased aggregate demand, GDP growth, and overall economic expansion. Conversely, when prices decline, the opposite effect often occurs, with reduced consumer confidence and companies in the real estate sector laying off workers, potentially triggering an economic downturn. Additionally, the growing trend of urbanization has led to increased demand for both rental and purchased housing. Consequently, there is a pressing need to develop more accurate methods for calculating house prices to reflect market conditions, informing decisions for investors, real estate professionals, financial institutions, and government agencies. Hence, the focus of this study.

3.2 Dataset collection

The dataset is a secondary data that was gotten from Kaggle. It is a housing price dataset and it consist of 50001 observations and 6 variables.

3.2.1 Dataset Attribute Information.

	Variables	Data Type	Meaning
1.	SquareFeet	Numeric	The area of the property in square feet.
2.	Bedrooms	Numeric	Number of bedrooms in the property.
3.	Bathrooms	Numeric	Number of bathrooms in the property
4.	Neighborhood	Categorical	location where the property is situated
5.	YearBuilt	Numeric	The year the property was constructed
6.	Price	Numeric	The price of the property

4. Objectives

The main objective of this study and selecting this dataset is to study the dataset, analyze the dataset and use it to develop effective and efficient house price prediction machine learning models that can help investors, realtors, banks, and government agencies make informed decision.

5. Features of the dataset.

The dataset is a supervised dataset. This is because it has dependent variable: price. It is also a regression task because the target variable (price) is numeric. Furthermore, it is a linear regression for the target variable is continuous. Based on these features, the models will be developed using regression machine learning algorithms.

6. Machine learning algorithm.

Machine learning algorithm is an algorithm that learns from historical data without being explicitly programed. There are so many regression machine learning algorithms. The following algorithms are randomly selected for this study.

- Randon Forest (RF)
- Support Vector Regression (SVR)
- XGBoost

6.1 Random Forest (RF).

According to IBM (2024), Random Forest is a machine learning algorithm which combines the output of multiple decision trees to reach a single result. It is used in both classification and regression task. It uses bagging ensemble method which involve training boostrap samples independently and then through averaging (for regression) and voting (for classification) make prediction. It has the following hyperparameters:

- N_estimators:

This parameter represents the number of trees in the forest. It's common to choose values that are large enough to ensure that the model has sufficient capacity to learn from the data and generalize well. However, too many trees can lead to overfitting and increased computational costs.

- Max_depth:

  This parameter controls the maximum depth of each tree in the forest. A deeper tree can capture more complex patterns in the data, but it also increases the risk of overfitting. Setting max_depth to None means that there is no limit on the depth of the trees, allowing them to grow until all leaves are pure or until all leaves contain less than min_samples_split samples.

- Min_samples_split:

This parameter sets the minimum number of samples required to split an internal node. Higher values prevent the tree from splitting nodes that have too few samples, which can help prevent overfitting.

- Min_samples_leaf:

This parameter sets the minimum number of samples required to be at a leaf node. Similar to min_samples_split, higher values can prevent overfitting by enforcing a minimum size for leaf nodes.

- Max_features:

This parameter determines the maximum number of features to consider when looking for the best split at each node in a decision tree.

The quality of node to split is determined using purity measurement method like Gini Index or entropy. A node with a smaller purity measurement value has more quality and more homogenous with the target variable. As a result, it is considered for splitting.

Mathematically,

$$Gini\ Index = 1 - \sum_{i=1}^{n} \left(P_i\right)^2$$

$$= 1 - [(P_+)^2 + (P_-)^2]$$ (Anshul Saini, 2022)

Where P_+ is the probability of a positive class and P_- is the probability of a negative class.

The final result is gotten by calculating weighted Gini Index.

- Bootstrap:

This parameter determines whether bootstrap samples are used when building trees. Bootstrap sampling helps introduce randomness into the training process, which can improve the diversity of individual trees and the overall performance of the Random Forest.

6.2. Support Vector Regression (SVR) Machine learning Algorithm.

According to Sethi (2024), support vector regression machine learning algorithm is an algorithm that makes prediction by finding a hyperplane that fits data points within a specified margin of error. It is used in regression task. It formulates objective functions using the features of the data point with constraints: to minimize deviation between predicted output and actual output while still respecting the margin of error. This objective function is then converted to optimization problem and solved using quadratic programming problem to fine the hyperplane. SVR has the following hyperparameters:

- Kernel function:

This parameter enables the algorithm to find the optimal hyperplane that fits the data points. There are different types of kernel function and the choice depends on the relationship between the features and the target variable. It includes Linear Kernel for linear relationship and Radial Basis Function (RBF) Kernel for non-linear relationship.

- Epsilon(ε):

This parameter defines the margin of tolerance around the regression line. It determines the size of the tube within which no penalty is associated with error. A large epsilon results in a wider tube, allowing more data points to reside within it, while a smaller epsilon creates a narrower tube, make the model more sensitive to deviation from the regression line.

- Regularization (C):

This parameter controls the trade-off between fitting training data allowing deviations within the margin of tolerance. A small value of C typically leads to larger deviations and potentially more error while a large value of C typically leads to small deviations and potentially fewer errors.

6.3. XGBOOST Machine Learning Algorithm

According to Guest_blog (2024), XGBoost machine learning algorithm is an algorithm that makes predictions by combining the decision of sequential models (decision trees). Each model is trained dependent on the mistake(error) the previous model made with the aim to correct the mistake or residual error which is the difference between predicted output and the actual output. It achieves this correction by minimizing the error through optimization of a specified loss function. XGBoost is used for both classification and regression problem and each objective parameter is associated with a loss function by default. For binary classification, the default loss function is logistic regression, this optimizes the log loss (binary cross- entropy). For regression task, the default loss function is reg: squarederror which optimizes the mean square error (MSE). It has the following hyperparameters:

- **learning_rate**:

The learning rate (also known as shrinkage or eta) controls the step size during the optimization process. A lower learning rate makes the model more robust to overfitting but requires more iterations to converge. Higher learning rates can lead to faster convergence but may cause the model to overshoot the optimal solution.

- **max_depth**:

The maximum depth of a tree controls the depth to which each tree in the ensemble is allowed to grow. Deeper trees can capture more complex patterns in the data but are more prone to overfitting. Setting a smaller **max_depth** can help prevent overfitting but may result in underfitting if the trees are too shallow to capture the underlying patterns.

- **n_estimators**:

The number of boosting rounds (or trees) to be built. Increasing the number of estimators generally improves model performance up to a certain point, as it allows the model to capture more complex patterns in the data. However, adding too many estimators can lead to overfitting and increased computational costs.

- **subsample**:

The fraction of training data to be randomly sampled (without replacement) for each boosting round. Subsampling can introduce randomness and help prevent overfitting by reducing the variance of the model. Using a value less than 1.0 (e.g., 0.8) can improve model generalization by training each tree on a different subset of the data.

- **colsample_bytree**:

The fraction of features to be randomly sampled (without replacement) for each tree. Similar to **subsample**, **colsample_bytree** introduces randomness and helps prevent overfitting by limiting the number of features considered for each tree. Using a value less than 1.0 (e.g., 0.8) can improve model generalization by training each tree on a different subset of features.

- **gamma**:

The minimum loss reduction required to make a further partition on a leaf node of the tree. A higher **gamma** value makes the algorithm more conservative, leading to fewer splits and potentially simpler trees. Increasing **gamma** can help prevent overfitting by penalizing overly complex trees.

- **reg_alpha and reg_lambda**:

L1 (Lasso) and L2 (Ridge) regularization terms applied to the leaf weights or the feature weights, respectively. These terms add penalties to the objective function to discourage overly complex models. Tuning **reg_alpha** and **reg_lambda** can help prevent overfitting and improve model generalization by controlling the complexity of the learned model.

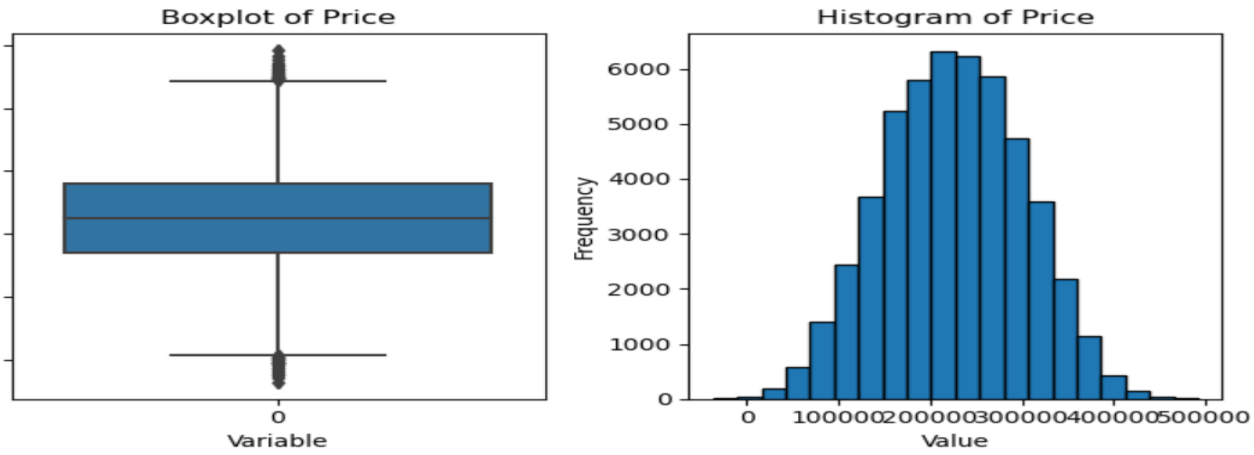
7. Dataset Importation.

This project was developed using python language and the development environment was Jupyter Notebook. The dataset was imported into the environment using pandas; a python library.

8. Data Cleaning.

Data cleaning is a process in machine learning modeling which focus on detecting and handling corrupt data such as wrong data type, duplicate rows, missing values and outliers to improve the quality of the dataset for effective and efficiency decision making. First, the dataset was examined for wrong data type, duplicate rows and missing values. From the examination, there was no wrong datatype, duplicate row and missing values. Then, each numerical variable was examined for outliers using box plot and histogram. From the examination, only variable price had outliers and it was handled as follows.

Fig.1



From the box plot, the presence of outliers can be seen beyond the box plot whisker's length. From the histogram, it was observed that some bins extended beyond zero (0) which indicate that some prices have negative sign and from domain knowledge, the price of a house cannot be negative. To handle this, absolute value of the price was taken.

Fig.2

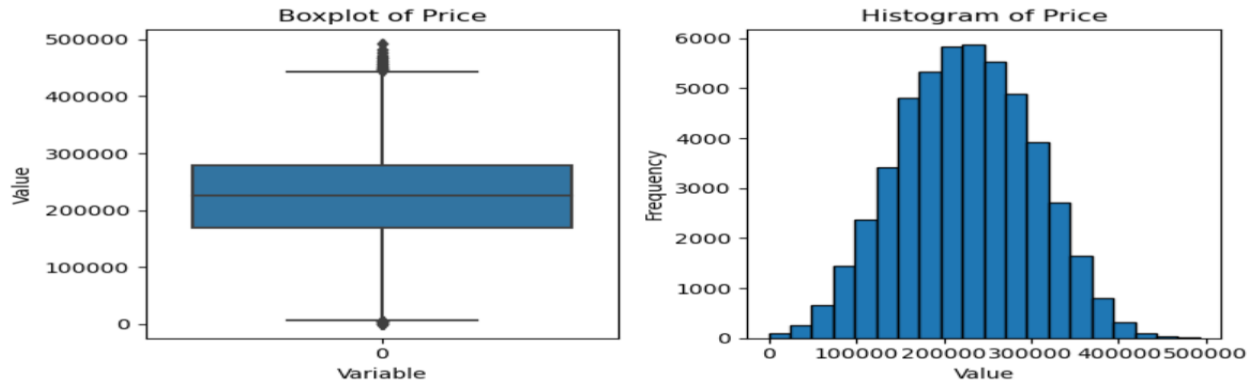


Figure 2 above shows the box plot and the histogram after the price absolute value was taken.

Then, the outliers were handled using winsorization method. During this phase, the lower and upper threshold were set at 1th and 99th percentile.

Fig.3

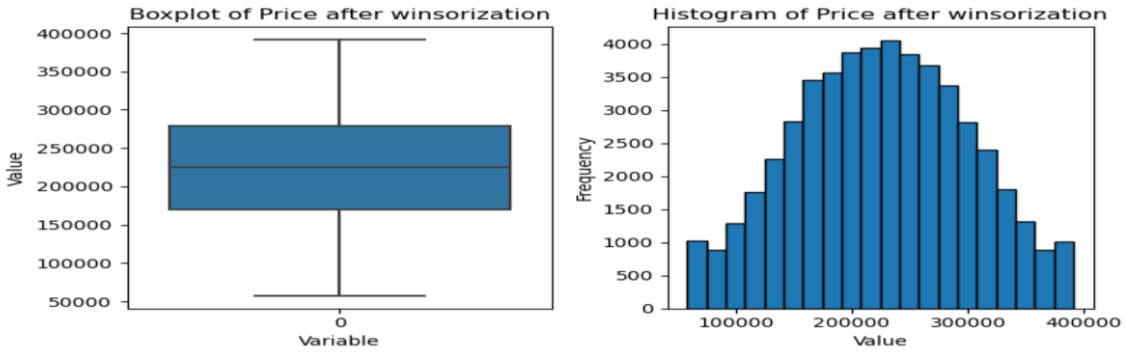
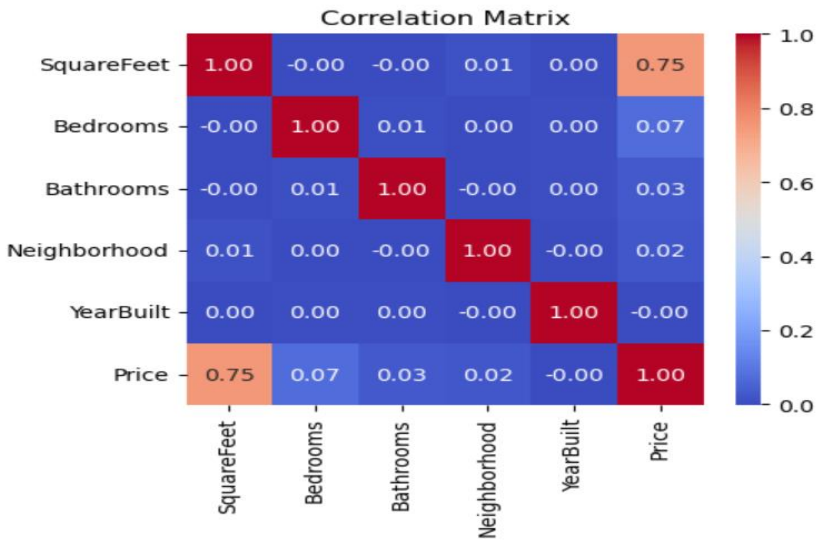


Figure 3 above shows the box plot and the histogram of the price column after winsorization.

9. Exploratory data analysis (EDA).

EDA is a process in machine learning modeling that aims at visualizing the distributions and relationships among the variables. In this project, the distribution of the variables was visualized using histogram when looking for the presence of outliers while the linear relationship among the variables was visualized using correlation matrix. This relationship is very important in machine learning modeling because it determines which variable that should be dropped or kept. If two variables have strong relationship; a relationship with correlation coefficient of -1 or +1 or very close to these values, one of the variables would be dropped. This is because training the model with the two variables will lead to collinearity. Under this condition, the model would not be able to differentiate their effect and this could lead to over fitting. This means that the model would be influenced by random fluctuation (occasional errors) and would not be able to generalize well on new and unseen data point. For the purpose of correlation plot, the categorical variables were label encoded, the correlation coefficient of the variable was calculated and the correlation matrix was plotted using heat map as shown below.

Fig.4



From the correlation matrix above, it was observed that none of the variable is strongly related to one another. So, all of them were used for the modeling.

After examining the linear relationship among the variables from the correlation matrix, the dataset was reimported. This was done because the categorical variables that are not ordinal were label encoded in order to calculate the correlation coefficient. So, the reimported dataset was cleaned using the same process, but the categorical variables were one hot encoded to form binary variable.

1O. Feature Engineering Analysis.

Feature engineering analysis is a phase in machine learning modeling that focus on the transformation of the original dataset for effective learning. In this study the following feature engineering analysis were carried out.

10.1. One hot encode.

The categorical variables were one hot encoded to form binary columns.

10.2. Row reduction.

Because of computational resources and time, the rows of the dataset were randomly reduced from 50,000 to 10,000.

After this phase, the processed dataset was saved to a new CSV file and was imported for modeling

11. Modeling:

11.1. Dataset split:

The dataset was split into three sets: 70% for training, 15% for validation and 15% for testing. Three machine learning models were developed using Support Vector Regression (SVR), Random Forest and XGBoost algorithms. They were trained, validated, cross - validated using k- fold and tested. Also, their learning graph was plotted to examine how they learn as the training progress.

11.2. Model Training:

In this phase, the algorithms were trained with 70% of the dataset to learn the relationship and underlying patterns among the variables and the data points.

11.3. Validation:

In this phase, 15% of the dataset and grid search method were used to tune the hyperparameters of the algorithms.

11.4. Cross - validation:

In this phase, the dataset was split into difference subset or folds using k-fold cross-validation method. This was done to examine how the model preforms across different subset.

11.5. Testing

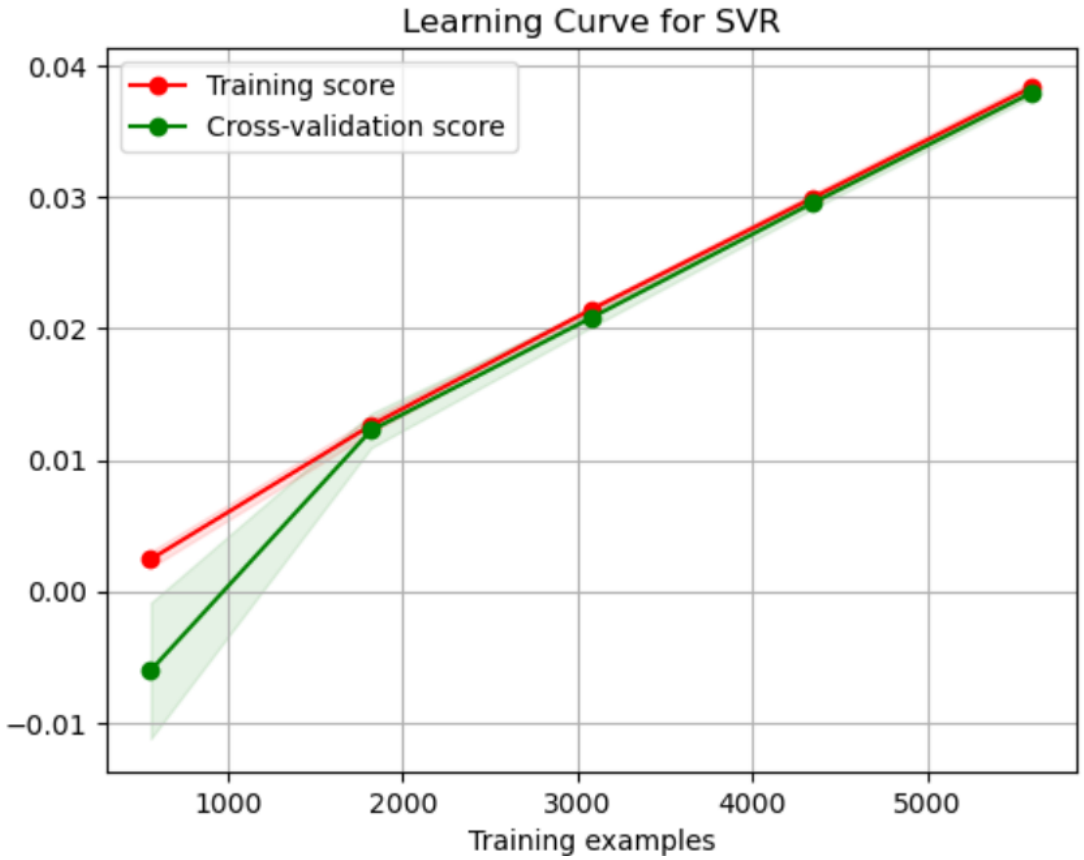
In this phase, the 15% of the dataset was used to test the model for this determines how it will perform or generalize on new and unseen data point.

During the development, each model had three configurations and the hyperparameters configuration with best performance was printed. The developed models and their result are shown below.

The Support Vector Regression results

Fig.5

```
Best parameters of SVR: {'C': 10, 'epsilon': 0.5, 'gamma': 0.1, 'kernel': 'rbf'}
Validation Set - Mean Absolute Error (MAE): 57816.73395095728
Validation Set - Mean Squared Error (MSE): 5030431499.21489
Validation Set - Coefficient of Determination (R^2): 0.043231267471325
Cross-Validation Mean Absolute Error (MAE): 60554.61
Cross-Validation Mean Squared Error (MSE): 5472507029.82
Cross-Validation Coefficient of Determination (R^2): 0.04
Training Set - Mean Absolute Error (MAE): 60524.881030581644
Training Set - Mean Squared Error (MSE): 5469579760.139359
Training Set - Coefficient of Determination (R^2): 0.0386621793616615
```

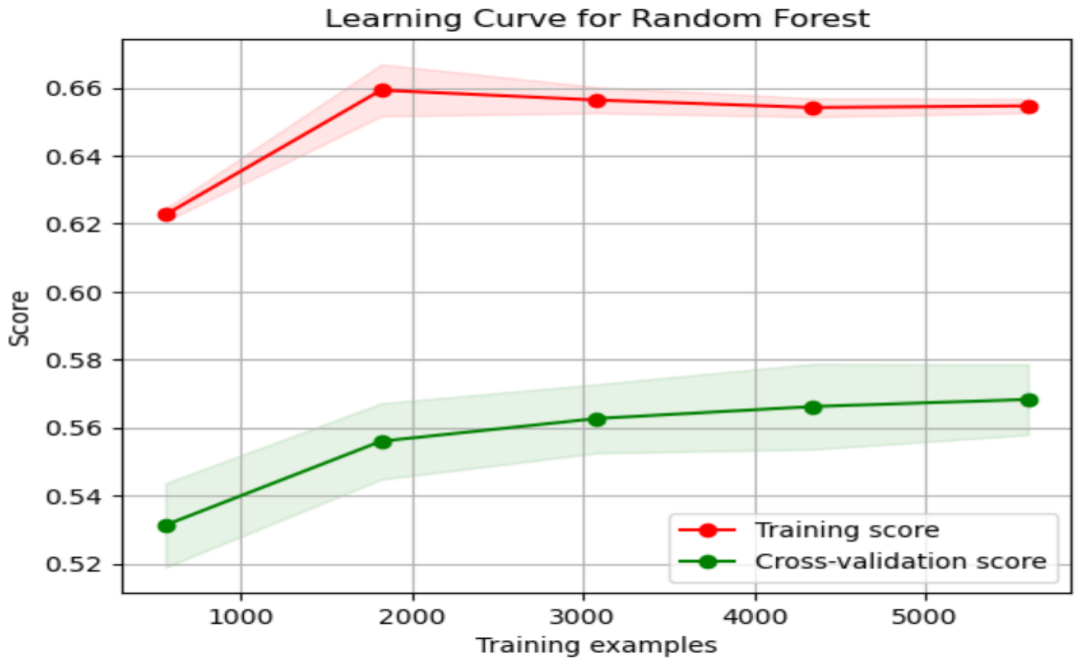



```
Test Set - Mean Absolute Error (MAE): 60284.88317326661
Test Set - Mean Squared Error (MSE): 5324250823.44652
Test Set - Coefficient of Determination (R^2): 0.040310127101546134
```

The Random Forest Results

Fig.6

```
Best Parameters of Random Forest: {'bootstrap': True, 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 8, 'min_samples_split': 2, 'n_estimato
rs': 300}
Validation Set - Mean Absolute Error (MAE): 40544.26499983374
Validation Set - Mean Squared Error (MSE): 2569572761.1546893
Validation Set - Coefficient of Determination (R^2): 0.511277139105486
Cross-Validation Mean Absolute Error (MAE): 39969.71
Cross-Validation Mean Squared Error (MSE): 2455773573.58
Cross-Validation Coefficient of Determination (R^2): 0.57
Training Set - Mean Absolute Error (MAE): 36551.86368450629
Training Set - Mean Squared Error (MSE): 2062609152.726223
Training Set - Coefficient of Determination (R^2): 0.6374741251309597
```

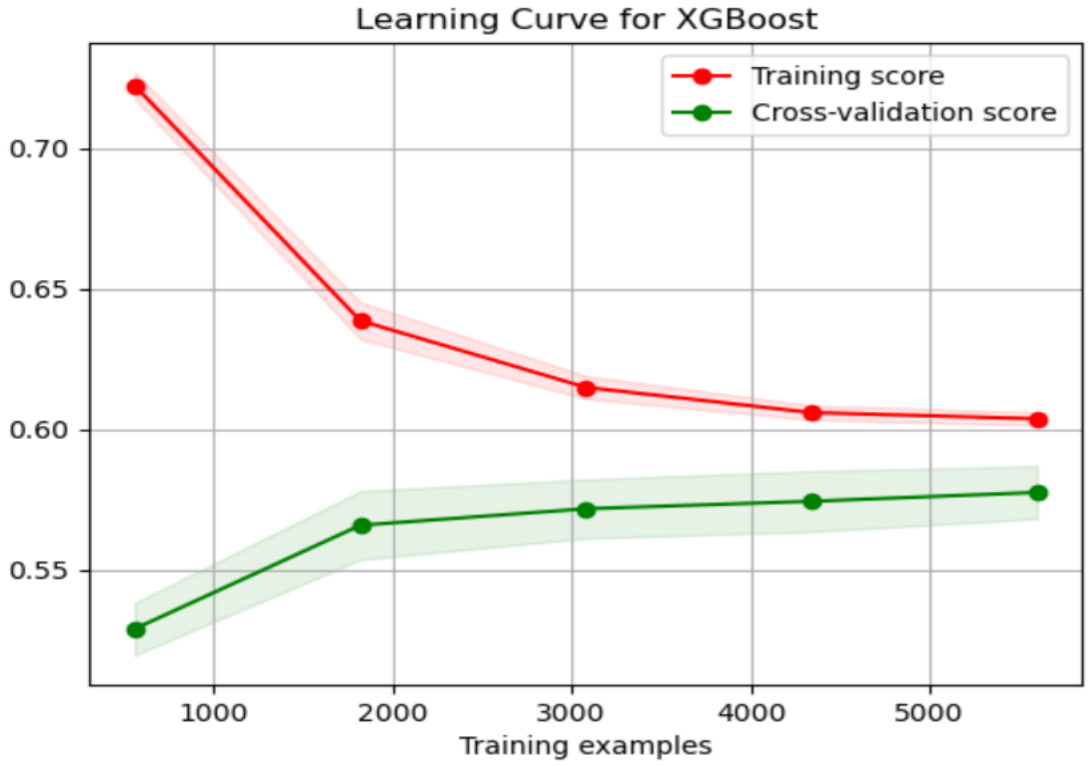



```
Test Set - Mean Absolute Error (MAE): 37775.517369300345
Test Set - Mean Squared Error (MSE): 2255574714.1439543
Test Set - Coefficient of Determination (R^2): 0.5934353428284687
```

The XGBOOST Results.

Fig.7

```
Best Parameters of XGBoost: {'colsample_bytree': 0.8, 'gamma': 0, 'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100, 'reg_alpha': 0.3, 'reg_lambd
a': 0, 'subsample': 1.0}
Validation Set - Mean Absolute Error (MAE): 39946.71830208333
Validation Set - Mean Squared Error (MSE): 2500849381.107564
Validation Set - Coefficient of Determination (R^2): 0.5243480617953262
Cross-Validation Mean Absolute Error (MAE): 39554.38
Cross-Validation Mean Squared Error (MSE): 2403530542.18
Cross-Validation Coefficient of Determination (R^2): 0.58
Training Set - Mean Absolute Error (MAE): 38574.38163950893
Training Set - Mean Squared Error (MSE): 2284883603.268745
Training Set - Coefficient of Determination (R^2): 0.5984069855628755
```




```
Test Set - Mean Absolute Error (MAE): 37392.866765625
Test Set - Mean Squared Error (MSE): 2211439098.7195115
Test Set - Coefficient of Determination (R^2): 0.6013907349693596
```

12. Discussion:

The Support Vector Regression

	Validation set	Cross validation set	Training set	Test set
MAE	57816.73	60554.61	60524.88	60284.88
MSE	5030431499.21	5472507029.82	54695579760.13	5324250823.44
R^2	0.0432	0.04	0.0386	0.04031

From the performance evaluation metrics above, it was seen that Support Vector Regression model did not perform well on the dataset as indicated by the relatively high MAE, MSE, and low R^2 values across all sets. Also, its R^2 Value of 0.04031 on the test set indicate that it can only explain 4% of the variance in the target variable. These suggest that the model might not be capturing the underlying patterns in the data effectively.

The Random Forest

	Validation set	Cross validation set	Training set	Test set
MAE	40544.26	39969.71	36551.56	37775.51
MSE	2569572761.15	2455773573.58	2062609152.72	2255574714.14
R^2	0.5112	0.57	0.6374	0.5112

The performance of the Random Forest model on the dataset seems to be good. its performance on training set, validation set and test set are similar, indicating that the model generalizes reasonably well to new and unseen data. Also, the Cross-Validation performance is similar to the Validation Set, suggesting consistency in model performance across different data splits. Additionally, its Coefficient of Determination (R^2) values are moderate with 0.593 on the test set, indicating that the model explains a reasonable amount, 59.3% of variance in the target variable.

The XGBOOST.

	Validation set	Cross validation set	Training set	Test set
MAE	39946.71	39554.38	38574.38	37392.80
MSE	2500849381.10	2403530542.18	2284883603.26	2211439098.71
R^2	0.5243	0.58	0.5984	0.6013

The XGBoost model performance on the dataset seems to be very good. Its better performance on the Test Set compared to the training and Validation Set is a good sign. Also, The Cross-Validation performance is generally close to the Validation Set, indicating that the model's performance is consistent across different splits of the data. Additionally, its Coefficient of Determination (R^2) values are reasonable across all the sets with 0.601 on the Test Set, indicating that the model explains a significant portion; 60.1% of the variance in the target variable.

13. Conclusion.

In summary, this research effectively tackled the prediction of housing prices through machine learning models (Support Vector Regression and XGBOOST), achieving Coefficients of Determination (R^2) of 0.593 and 0.601, *MAE value of 37775.51* and *37392.80 and MSE value of 2255574714.14 and* 2211439098.71 on the Test Set respectively. Despite acknowledging certain limitations, the results offer valuable insights for investors, real estate professionals, financial institutions, and government bodies. Furthermore, it presents opportunities for future investigation. Subsequent studies could delve into advanced algorithms, integrate more features, or gather a broader and more diverse dataset, potentially enhancing the model's accuracy and applicability.

14. Reflection on professional, ethical, and legal issues in machine learning and house price dataset.

In terms of professional standard, I ensured that this study followed data science processes; which are Problem identification, Raw dataset collection, Clean/pre-processing of raw dataset, Exploratory data analysis, Building model and Analyze results and Presentation of results in a visual way.

In terms of ethical concerns, I considered transparency. I ensured that the source of the dataset, how it was cleaned, processed and split to develop the models were stated. Also, I considered Data Integrity and Accuracy. So, I thought of potential source of errors or biases like missing values, wrong data type, duplicate rows and outliers and ensued that the dataset was examined and handled appropriately for these issues. In addition, I considered Continuous Improvement and so recommended the work for further study which could involve exploring more advanced algorithms, incorporating additional features, or collecting an expanded and varied dataset to improve the model’s performance.

References:

Ahtesham, M., Bawany, N. Z., & Fatima, K. (2020). *House Price Prediction using Machine Learning Algorithm - The Case of Karachi City, Pakistan.* https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9300074

Collins English Dictionary. (n.d.). *House prices. In Collins English Dictionary. Retrieved Mach 30th, 2024.* https://www.collinsdictionary.com/dictionary/english/house-prices#google_vignette

Guest-Blog. (2024). XGBoost: *Introduction to XGBoost Algorithm in Machine Learning.* https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/

Hoa, W. K. O., Tang, B. S., & Wong, S. W. (Year*). Predicting property prices with machine learning algorithms. Journal of Property Research, 38(1)*, 48–70. https://doi.org/10.1080/09599916.2020.1832558

IBM (2024). *What is random forest?* https://www.ibm.com/topics/random-forest

Jha, S. B., Babiceanu, R. F., Pandey, V., & Jha, R. K. (2006). *Housing Market Prediction Problem using Different Machine Learning Algorithms: A Case Study*. https://arxiv.org/abs/2006.10092

Park, B., & Bae, J. K. (2015). *Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data*. http://tinyurl.com/yeydkwf8

Ravikumar, A. S. (2018) *Real Estate Price Prediction Using Machine Learning*. https://norma.ncirl.ie/3096/1/aswinsivamravikumar.pdf

Sethi.A.(2022). *Support Vector Regression Tutorial for Machine Learning.* https://www.analyticsvidhya.com/blog/2020/03/support-vector-regression-tutorial-for-machine-learning/

Truong, Q., Nguyen, M., Dang, H., & Mei, B. (2019). *Housing Price Prediction via Improved Machine Learning Techniques. In 2019 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI2019).* http://tinyurl.com/bdh5a4sr

Zulkifley, N. H., Rahman, S. A., Ubaidullah, N. H., & Ibrahim, I. (2020*). House Price Prediction using a Machine Learning Model: A Survey of Literature. I.J. Modern Education and Computer Science, 6, 46–54*. https://doi.org/10.5815/ijmecs.2020.06.04