Contents

[1. Introduction 3](#_Toc149832230)

[2. Data Summary 4](#_Toc149832231)

[3. Literature Review 4](#_Toc149832232)

[A. Regression 4](#_Toc149832233)

[A1. Linear regression: 4](#_Toc149832234)

[A2. WoE Binning: 4](#_Toc149832235)

[A3. Random Forest in Spatial Data Analysis: 4](#_Toc149832236)

[A4. Gradient Boosting in Spatial Data Analysis: 5](#_Toc149832237)

[A5. Artificial Neural Network 5](#_Toc149832238)

[4. Methods and Transformation, and Results 6](#_Toc149832239)

[A. Regression 6](#_Toc149832240)

[Feature engineering 6](#_Toc149832241)

[Model making 8](#_Toc149832242)

[Hybrid model. 8](#_Toc149832243)

[Performance Comparison and Interpretation: 9](#_Toc149832244)

[Results 10](#_Toc149832245)

[B. Classification 11](#_Toc149832246)

[B.1. SVM 11](#_Toc149832247)

[B.2. ANN 11](#_Toc149832248)

[B.3. KNN 11](#_Toc149832249)

[B.4. Decision Tree 11](#_Toc149832250)

[C. Time series 11](#_Toc149832251)

[D. Bayesian Network 11](#_Toc149832252)

[E. Naïve Bayesian Network 11](#_Toc149832253)

[F. Clustering 11](#_Toc149832254)

[5. Summary of Results 11](#_Toc149832255)

[6. References 11](#_Toc149832256)

2.1. District Split

Haiyu part, to mentioned and provide the map how he split the map, how it helps us in the further report

2.2. Discretizing data for Model   
  
 2.3. Data Overview

-some histograms, boxplot to explain the data itself   
 - relationship heatmap (categorical, numerical)

2.4. Target variable   
 content would be: price->price\_by\_area -> log( price\_by\_area)  
 interms of ‘unit’ categroy does not have building holder right......we decide to take ‘price\_by\_area’ instead  
 no normality so -> log(price\_by\_area)

2.5. Predictor variable.  
 content be like, we separate into three dataset such as house, unit, and townhouse, why we separate it, what is the purpose and reason

2.6. Model Prediction  
 we split 70% and 30% consistently in the report to train the training data and test data

1. Literature Review  
    - contents like what the model doing, the purpose, the reason doing, what the input and output would look like, the framework

4. Methods and Transformation, and Results  
 -Every part of model results, and explain the meaning, providing diagram, and accuracy

Figures

[Figure 1 Correlation plot for regression 7](#_Toc149750970)

[Figure 2 Performance of hybrid model for different weights of ensemble models 10](#_Toc149750971)

Tables

[Table 1 Coefficients and P values of features in linear regression 8](#_Toc149750975)

[Table 2 Importance level of all features 8](#_Toc149750976)

[Table 3 Performance parameters of ensemble regression and hybrid model 10](#_Toc149750977)

[Table 4 Relative importance of all features for Hybrid regression model 11](#_Toc149750978)

# 1. Introduction

# 2. Data Summary

# 3. Literature Review

## A. Regression

### A1. Linear regression:

Linear regression, often simply termed as 'regression', predicts a continuous outcome variable based on one or more predictor variables. Its primary measure of feature importance is derived from the coefficients of the predictors in the regression equation.

The magnitude and sign of the coefficients indicate the direction and strength of the relationship between the predictor and the outcome. However, because these coefficients are scale-dependent, comparing them directly can be misleading, especially when predictors are of different scales. To address this, standardized coefficients (often termed 'beta coefficients') can be used, which represent the change in the outcome for a one standard deviation change in the predictor, making them more comparable [1].

The p-values associated with each coefficient tell us if the predictor is statistically significant. While this doesn't directly measure importance, a non-significant predictor is generally considered not to be an important predictor in the context of the other variables in the model ( [2].

### A2. WoE Binning:

An essential pre-processing step often integrated into this workflow is the Weight of Evidence (WoE) method. Conceived as a binning transformation, WoE transmutes continuous predictors into categorical bins, attributing a WoE value to each [3]. This maneuver standardizes predictors, rendering them apt for the multinomial logistic regression framework. Each bin embodies observations presumed to share a consistent relationship with the target. The WoE formula is articulated as:

This transformation intrinsically recalibrates variables to a logistic scale. A notable characteristic of WoE is its zero value when one price category's distribution parallels another's, suggesting the bin's potential irrelevance ( [3]. Such bins might be merged or excised from the analysis.

### A3. Random Forest in Spatial Data Analysis:

Random Forest™, a trademark of [4], is an ensemble learning method that constructs multiple decision trees at training time and produces the mode of the class outputs for classification or the mean prediction for regression. In spatial data analysis, Random Forest has been recognized for its robustness in handling large datasets with missing values, making it apt for applications where data completeness is a challenge.

While the method is not always the go-to technique in geospatial literature, several researchers acknowledge its potential. For instance, [5] highlighted the suitability of Random Forest in mineral prospectivity mapping, noting its capability to handle missing values seamlessly. Moreover, the adaptability of the Random Forest in integrating distances to geological features as predictors demonstrates its flexibility in spatial contexts.

### A4. Gradient Boosting in Spatial Data Analysis:

Gradient Boosting is another ensemble technique, but unlike Random Forest, it constructs predictive models in a stage-wise manner, with each subsequent model being built to correct the errors of its predecessor. This iterative improvement typically results in a more accurate and generalized model, especially with spatial data that often contains non-linear and complex relationships.

The application of Gradient Boosting in spatial data analysis has gained traction in recent years. For instance, spatial autocorrelation, a common phenomenon in geospatial datasets, can lead to overestimation or underestimation in conventional statistical models. Gradient Boosting, with its iterative correction mechanism, can potentially mitigate the effects of spatial autocorrelation [6] Additionally, in land cover classification tasks, Gradient Boosting has been found to outperform traditional classification algorithms, particularly in areas with intricate land cover transitions [7]

Furthermore, the integration of geospatial features, such as distances to specific landmarks or geographical features, can be intuitively incorporated into the Gradient Boosting framework, providing nuanced insights into spatial relationships.

### A5. Artificial Neural Network

When it comes to house price prediction, artificial neural networks (ANN) have shown quite compelling potential as a machine learning tool. Unlike Gradient Boosting, ANN is a neuron- and hierarchical-based model that makes predictions by learning complex patterns in data.

Research shows that ANN has achieved remarkable results in house price prediction. In predicting the real estate market, the ANN model shows high prediction accuracy. It can process many input features, such as geographical location, house area, surrounding facilities, etc., thereby more accurately capturing the changing patterns of housing prices. Research also points out that ANN performs better than traditional linear models in dealing with nonlinear and complex housing price change patterns [8].

In addition, the application of ANN is not limited to data processing but can also be combined with geospatial features to improve the accuracy of predictions. By incorporating geographical location factors, surrounding environment, and housing characteristics into the ANN model, a more comprehensive housing price prediction can be achieved. This method of comprehensively considering spatial information has shown high feasibility and accuracy in house price prediction tasks [9].

Overall, ANN, as a powerful machine learning tool, can predict housing prices more accurately, and by integrating geospatial features, the prediction results are more convincing and interpretable.

### A6. Support Vector Machine

Support Vector Machine (SVM), introduced by Vapnik and Cortes in 1995 [10], is a supervised learning algorithm that has proven to be effective in both classification and regression tasks. It works by finding the optimal hyperplane that maximizes the margin between the closest data points (support vectors) of different classes while minimizing the classification error. SVM is particularly useful when dealing with non-linear and complex relationships in data, as it can employ kernel functions to map data into higher-dimensional spaces where linear separation is possible.

In the context of housing price prediction, SVM has gained recognition as an effective tool for modeling and forecasting property values. A study applied SVM to predict housing prices by considering various features such as property size, location, number of bedrooms, and amenities [11]. The study demonstrated the capability of SVM to handle regression tasks effectively, showing competitive performance compared to other regression methods.

There are several reasons why SVM is a suitable choice for predicting housing prices. Firstly, SVM can handle non-linear relationships and complex interactions between features, which are common in real estate markets. Secondly, it allows the integration of geographical and spatial features, such as distance to schools, parks, and public transport, which can significantly impact property values. Finally, SVM's ability to provide a margin of error in its predictions allows for a nuanced understanding of housing price fluctuations, which is valuable for real estate professionals and investors.

# 4. Methods and Transformation, and Results

## A. Regression

### Feature engineering

We shall begin making regression models by first analyzing correlation in-between the target and predictor variables.

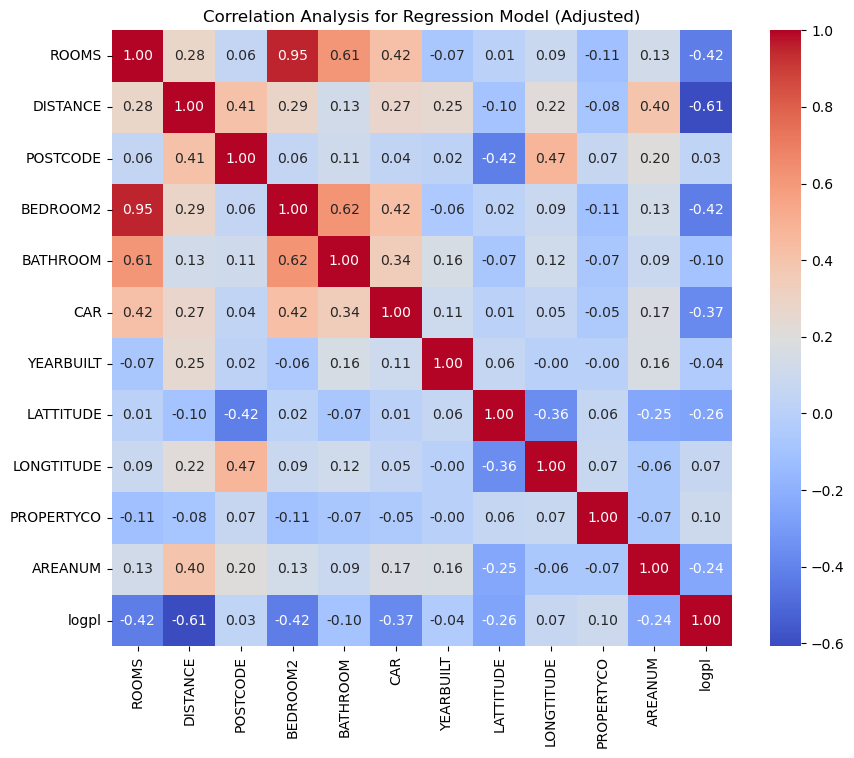


Figure 1 Correlation plot for regression

Here, *ROOMS, DISTANCE, BEDROOM2, CAR, LATITUDE AND AREANUM* show good correlation with our target variable.

In addition to that variables *BEDROOM2* and *ROOMS*, *LONGITUDE* and *POSTCODE*, *ROOMS* and *CAR* shows strong collinearity among themselves.

A linear regression was fitted to the dataset to find significance of variables. The regression model yielded an R^2 of 0.61. In cross validation the mean MSE was 0.232 and standard deviation was 0.018.

The significance of the numerical variables is as follows.

|  |  |  |
| --- | --- | --- |
|  | **Coefficient** | **P-Value** |
| **const** | -131.834 | 1.05E-36 |
| **ROOMS** | -0.13603 | 1.02E-10 |
| **DISTANCE** | -0.08367 | 0 |
| **POSTCODE** | 0.001649 | 8.62E-69 |
| **BEDROOM2** | -0.08192 | 0.000103 |
| **BATHROOM** | 0.159209 | 1.34E-40 |
| **CAR** | -0.1175 | 4.33E-54 |
| **YEARBUILT** | 0.002123 | 1.08E-31 |
| **LATTITUDE** | -2.29499 | 9.9E-119 |
| **LONGTITUDE** | 0.313022 | 2.89E-05 |
| **PROPERTYCO** | 1.59E-06 | 0.273303 |
| **AREANUM** | -0.02371 | 5.35E-11 |

Table 1 Coefficients and P values of features in linear regression

As seen in the table above, all features except PROPERTYCO are significant so we discard PROPERTYCO for all further analysis. So, we proceed further with the remaining features and check their importance level for a random forest regressor.

|  |  |  |
| --- | --- | --- |
|  | Feature | Importance level |
| 1 | DISTANCE | 0.363249 |
| 2 | type\_u | 0.279393 |
| 3 | LATTITUDE | 0.075628 |
| 4 | LONGTITUDE | 0.071703 |
| 5 | YEARBUILT | 0.067997 |
| 6 | type\_h | 0.049694 |
| 7 | POSTCODE | 0.030542 |
| 8 | type\_t | 0.020609 |
| 9 | CAR | 0.01252 |
| 10 | ROOMS | 0.008343 |
| 11 | BEDROOM2 | 0.008307 |
| 12 | BATHROOM | 0.007938 |
| 13 | AREANUM | 0.004077 |

Table 2 Importance level of all features

For further analysis using ensemble methods, from the above Features rather than defining an arbitrary threshold and choosing some top features, we chose all features to get the best model possible. However, we will discard *ROOMS*, *POSTCODE* and *AREANUM* as they have high collinearity with *BEDROOM2*, *LONGITUDE* and *DISTANCE*.

### Model making

**Random Forest model.**

After trying multiple parameters, the best model was obtained by splitting the dataset into 30% for testing and 70% for training. The number of trees in the forest was limited to 300 and other hyperparameters like max\_depth and min\_samples\_split is not explicitly defined, so they will be set to their default values as per the Scikit-learn library. The R^2 obtained for the fitted Random Forest model is 0.826.

**Gradient boosting model.**

The number of boosting stages (trees) to be run. essentially the number of sequential trees being modeled was limited to 300 and other hyperparameters like max\_depth and learning\_rate is not explicitly defined, so they will be set to their default values as per the Scikit-learn library. The R^2 obtained for the fitted Gradient boosting model is 0.829.

**Cross validation:**

The result obtained for 5-fold cross validation for both random forest and Gradient boosting is as follows.  
For the Random Forest:

* Mean MSE: ≈0.118
* Standard Deviation MSE: ≈0.012

For the Gradient Boosting:

* Mean MSE: ≈0.114
* Standard Deviation MSE: ≈0.024

Model Performance: The Gradient Boosting model has a slightly lower mean MSE compared to the Random Forest, suggesting that it has a better average performance in terms of prediction error.

Model Consistency: The Random Forest model has a lower standard deviation in MSE, indicating that its performance is more consistent across different folds of the data. The Gradient Boosting model, while performing better on average, shows more variability in its performance.

Combining both models might yield better performance.

### Hybrid model.

Here we have combined Random Forest and Gradient boosting models with weighted averaging method.

Below given graph shows different levels of R^2(y-axis) vs weight of the Random Forest(x-axis).

A graph of a weight loss

Description automatically generated with medium confidence

Figure 2 Performance of hybrid model for different weights of ensemble models

Using the above graph, we chose 0.45 as the weight for Random Forest and 0.55 as the weight for Gradient boosting.

The final hybrid model yielded an R^2 of 0.837.

By doing 5-fold Cross validation on the hybrid model we obtained the following.

* Mean MSE: ≈0.099
* SD MSE: ≈0.011

### Performance Comparison and Interpretation:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Random Forest | Gradient Boosting | Hybrid Model |
| MSE | 0.102 | 0.100 | 0.096 |
| MAE | 0.215 | 0.218 | 0.210 |
| R^2 | 0.826 | 0.829 | 0.837 |
| 10 fold CV Mean MSE | 0.118 | 0.114 | 0.010 |
| 10 fold CV SD MSE | 0.012 | 0.024 | 0.013 |

Table 3 Performance parameters of ensemble regression and hybrid model

* Performance: The Weighted Average Hybrid Model shows a significant improvement over the initial models in terms of the Mean MSE. It has the lowest Mean MSE among the three, indicating that, on average, it predicts with less error.
* Consistency: The Random Forest Regressor is the most consistent model with the lowest SD in MSE. However, the Weighted Average Hybrid Model also shows good consistency, with its SD in MSE being close to that of the Random Forest and lower than that of the Gradient Boosting.
* Error Reduction: The hybrid model's improvement in Mean MSE suggests that combining the models helps reduce prediction error, potentially by leveraging the strengths of both individual models.
* Stability: Despite the Gradient Boosting Regressor's higher variability (SD in MSE), its strengths when combined with the Random Forest in the hybrid model led to a more stable model than Gradient Boosting alone.

In summary, the Weighted Average Hybrid Model appears to offer a favorable balance between error minimization and consistency, outperforming the individual initial models in terms of the Mean MSE while maintaining a reasonable level of variability across different data subsets. In addition to that the Hybrid model also yields an R^2 value higher than both Individual models.

### Results

First, we shall check Importance of features in predicting the variance in target variable.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Feature | Random Forest | Gradient\_boosting | Mean\_Importance |
| 1 | DISTANCE | 0.359768 | 0.383445 | 0.371606 |
| 2 | type\_u | 0.281138 | 0.197066 | 0.239102 |
| 3 | type\_h | 0.050886 | 0.166818 | 0.108852 |
| 4 | LATTITUDE | 0.094765 | 0.100073 | 0.097419 |
| 5 | LONGTITUDE | 0.089964 | 0.074995 | 0.082479 |
| 6 | YEARBUILT | 0.06869 | 0.041698 | 0.055194 |
| 7 | type\_t | 0.017458 | 0.019362 | 0.01841 |
| 8 | CAR | 0.014023 | 0.007364 | 0.010693 |
| 9 | BEDROOM2 | 0.014529 | 0.004034 | 0.009282 |
| 10 | BATHROOM | 0.008779 | 0.005145 | 0.006962 |

Table 4 Relative importance of all features for Hybrid regression model

* DISTANCE: This is the most important feature in predicting the log-transformed price (logpl), with the highest mean importance across both models. It suggests that the distance from the city center is a crucial factor in determining property prices.
* Property Type (type\_u, type\_h, type\_t): The type of property (unit, house, townhouse) also plays a significant role, with 'type\_u' (unit) having a notably high importance, especially in the Random Forest model. This indicates that the type of property is a significant factor in its price.
* Geographic Location (LATTITUDE, LONGTITUDE): Latitude and longitude, which represent the property's geographic location, are also important predictors. This aligns with the common real estate mantra of "location, location, location."
* YEARBUILT: The year in which the property was built holds some importance, suggesting that newer or older homes might have specific price implications.
* CAR, BEDROOM2, BATHROOM: These features have lower importance in predicting property prices. However, they still contribute to the overall predictive power of the model.

Based on these insights, we can derive the following results:

Proximity to City Center: Properties closer to the city center are likely to be more expensive. Real estate agents and developers should consider the distance from city centers as a key factor when valuing properties or planning new developments.

Property Type: Property being a unit tends to have a significant impact on price, possibly due to their popularity or availability in certain areas.

Location-Specific Marketing: The importance of latitude and longitude suggests that certain locations within Melbourne are more sought after.

New Developments and Renovations: The year a property was built affects its price. This shows that newer properties are preferred more in comparison to older ones.

Feature Additions and Renovations: Features like the amount of cars that can be parked have more impact on property prices than number of bedrooms, which is surprising.

These insights can guide decision-making in the real estate market, from individual property valuations to broader market strategies.​​

## B. Classification

### B.1. SVM

**Features Engineering**

First, the correlation between variables is analyzed and variables with significant correlation are removed, thereby reducing the complexity of the model, and avoiding possible multicollinearity problems.

A screenshot of a graph

Description automatically generated

Figure 3 Correlation Heatmap for Categorical Variables for Different Types of Buildings

The chart above illustrates the correlation among categorical variables for three different types of buildings. While there are variations in the specific numerical values, it is evident that there is a significant correlation between SUBURB and REGIONNAME, as well as between COUNCILARE and POSTCODE, with all correlations exceeding 95%. Due to the comparatively lower number of distinct values in REGIONNAME, these variables have been reduced to just REGIONNAME for further analysis.

Next, the correlation between numerical variables is analyzed.

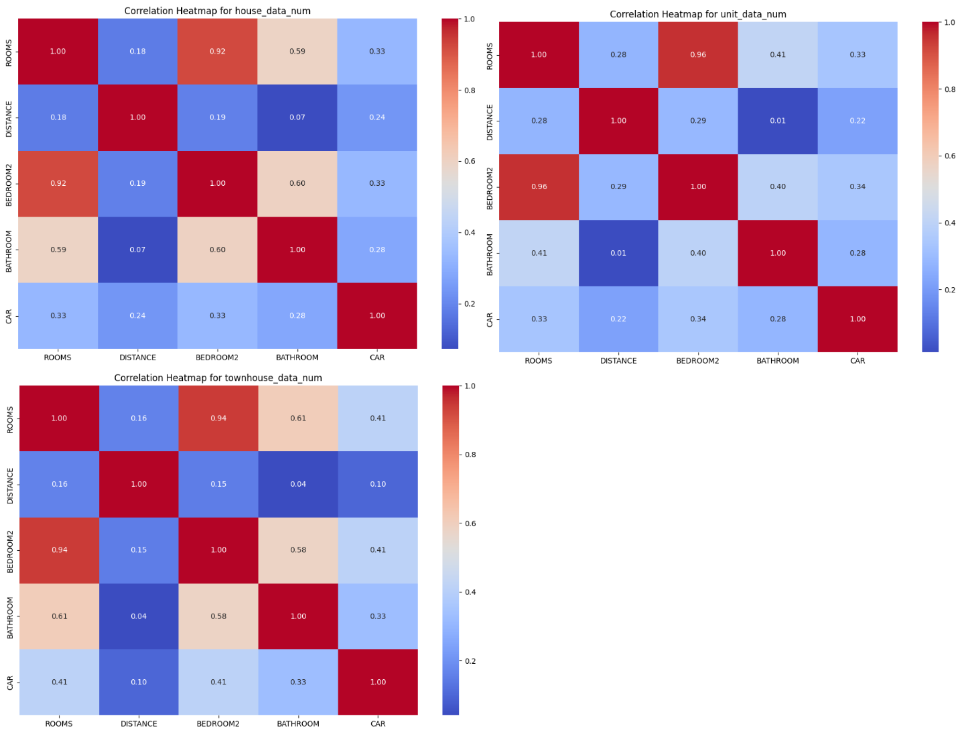


Figure 3 Correlation Heatmap for Numeric Variables for Different Types of Buildings

From the visualizations, it is evident that the numerical variables for houses and townhouses exhibit a consistent pattern of correlation. Both Bedroom2 and Bathroom show a significant relationship with Rooms, exceeding 60%. However, in the case of units, the correlation between Bathroom and Rooms is only 41%, indicating a less pronounced relationship. Based on these correlations, it is advisable to remove the Bedroom2 and Bathroom variables for houses and townhouses, while retaining the Bathroom variable for units.

**Model Making**

After experimenting with various parameter combinations, the model that achieved the highest accuracy for house data is characterized by the following parameter settings: a regularization parameter (C) of 10, the utilization of the Radial Basis Function (RBF) kernel (commonly known as the Gaussian kernel, which is suitable for non-linear classification problems), and a gamma value of 0.01.

The performance of this model is as follows:

|  |  |  |
| --- | --- | --- |
|  | Metric | Value |
| 0 | Accuracy | 0.71 |
| 1 | Precision | 0.71 |
| 2 | Recall | 0.71 |
| 3 | F1 Score | 0.60 |

Figure 5 Performance Metrics for house’s svm model

In the context of Unit Data, various parameter combinations were explored. Among these, two specific combinations emerged, both achieving an accuracy of 0.82. These combinations are as follows: {'C': 1.0, 'kernel': 'rbf', 'gamma': 0.1, 'random\_state': 42'} and {'C': 10.0, 'kernel': 'rbf', 'gamma': 0.01, 'random\_state': 42'}.

Adhering to Occam's razor principle, we lean towards selecting the simpler model. Simplicity often leads to improved generalization and reduced risk of overfitting the training data. Consequently, our preference is to opt for the parameter combination {'C': 1.0, 'kernel': 'rbf', 'gamma': 0.1, 'random\_state': 42'}. This choice is underpinned by the fact that this model exhibits a more straightforward decision boundary, characterized by a lower 'C' value and a higher 'Gamma' value.The performance of this model is as follows:

|  |  |  |
| --- | --- | --- |
|  | Metric | Value |
| 0 | Accuracy | 0.82 |
| 1 | Precision | 0.73 |
| 2 | Recall | 0.82 |
| 3 | F1 Score | 0.76 |

Figure 6 Performance Metrics for unit’s svm model

Likewise, when it comes to the townhouse data, a range of parameter models were employed to train the SVM model. After assessing their performance, it became evident that three parameter combinations yielded an accuracy of 79%. These parameter sets are as follows: {'C': 0.1, 'kernel': 'rbf', 'gamma': 'scale', 'random\_state': 42'}, {'C': 1.0, 'kernel': 'rbf', 'gamma': 'scale', 'random\_state': 42'}, and {'C': 1.0, 'kernel': 'rbf', 'gamma': 0.001, 'random\_state': 42'}.

In line with the Occam's razor principle, we opt for the model with the parameters {'C': 1.0, 'kernel': 'rbf', 'gamma': 0.001, 'random\_state': 42'}. This choice is made because it features a higher 'C' value and a smaller 'gamma' value. This configuration implies a reduced penalty for misclassification and a relatively uncomplicated decision boundary, aligning with the principle of favoring simplicity in model selection.The performance of this model is as follows:

|  |  |  |
| --- | --- | --- |
|  | Metric | Value |
| 0 | Accuracy | 0.82 |
| 1 | Precision | 0.73 |
| 2 | Recall | 0.82 |
| 3 | F1 Score | 0.76 |

Figure 7 Performance Metrics for townhouse’s svm model

### B.2. ANN

The model for house data with the highest accuracy was achieved by experimenting with a range of parameter combinations and it includes three hidden layers with 64, 32, and 16 neurons, the model was trained for 20 epochs, and each epoch used a batch size of 32 samples.

|  |  |  |
| --- | --- | --- |
|  | Metric | Value |
| 0 | Accuracy | 0.62 |
| 1 | Precision | 0.62 |
| 2 | Recall | 0.62 |
| 3 | F1 Score | 0.62 |

Figure 7 Performance Metrics for house’s ann model

The model for unit data with the highest accuracy was also achieved by experimenting with a range of parameter combinations. In the end, it was found that when the ANN model has 128 neurons in the first hidden layer, 64 neurons in the second hidden layer, 10 training epochs, and a batch size of 64, the model achieved the highest accuracy, reaching 0.76.

|  |  |  |
| --- | --- | --- |
|  | Metric | Value |
| 0 | Accuracy | 0.76 |
| 1 | Precision | 0.69 |
| 2 | Recall | 0.73 |
| 3 | F1 Score | 0.71 |

Figure 8 Performance Metrics for unit’s ann model

For the townhouse ANN model, experimenting with different parameter combinations, the model achieves the highest accuracy when the first hidden layer has 128 neurons, the second hidden layer has 64 neurons, the number of training epochs is set to 10, and the batch size for each training step is 64.

|  |  |  |
| --- | --- | --- |
|  | Metric | Value |
| 0 | Accuracy | 0.77 |
| 1 | Precision | 0.67 |
| 2 | Recall | 0.79 |
| 3 | F1 Score | 0.71 |

Figure 9 Performance Metrics for unit’s ann model

### B.3. KNN

### B.4. Decision Tree

## C. Time series

## D. Bayesian Network

## E. Naïve Bayesian Network

## F. Clustering

# 5. Summary of Results

# 6. References

|  |  |
| --- | --- |
| [1] | L. E. Toothaker, L. S. Aiken and S. G. West, "Multiple Regression: Testing and Interpreting Interactions," *The Journal of the Operational Research Society,* p. p.119, 1994. |
| [2] | N. R. a. Draper and H. Smith, Applied regression analysis, Hoboken, New Jersey : John Wiley & Sons, Inc., 1998. |
| [3] | H. SCHAEBEN and G. SEMMLER, "The quest for conditional independence in prospectivity modeling： weights-of-evidence, boost weights-of-evidence, and logistic regression," *Frontiers of earth science,* pp. Vol.10 (3), p.389-408, 2016. |
| [4] | L. Breiman, "Random forests," *Machine learning,* pp. Vol.45 (1), p.5-32, 2001. |
| [5] | E. J. M. Carranza and A. G. Laborte, "Random forest predictive modeling of mineral prospectivity with small number of prospects and data with missing values in Abra (Philippines)," *Computers & geosciences,* pp. Vol.74, p.60-70, 2015. |
| [6] | J. Elith and J. Leathwick, "Boosted Regression Trees for ecological modeling," 12 June 2011. [Online]. Available: https://cran. r-project. org/web/packages/dismo/vignettes/brt. pdf. |
| [7] | P. O. Gislason, J. A. Benediktsson and J. R. Sveinsson, "Random Forests for land cover classification," *Pattern recognition letters,* pp. Vol.27 (4), p.294-300, 2006. |
| [8] | N.-T. . M. Julia, C. M. José and R. J. Francisco, "Artificial Neural Networks for Predicting Real Estate Prices," *Revista de métodos cuantitativos para la economía y la empresa,* vol. 15, no. 1, pp. 29-44, 2013. |
| [9] | X. Xiaojie and Z. Yun, "House price forecasting with neural networks," *Intelligent systems with applications,* vol. 12, p. 200052, 2021. |
| [10] | C. Corinna and V. Vladimir, "Support-vector networks," *Machine Learning,* vol. 20, p. 273–297 , 1995. |
| [11] | W. Y. Jiao, "Housing Price prediction Using Support Vector Regression," *Master's Projects,* p. 540, 2017. |