ALY 6020 – CRN 70409

Predictive Analytics

Professor: Amin Karimpour Northeastern University, Boston



Module 5 Assignment - Investing in Nashville

Submitted by – Santhi Kiran Chavali

# Introduction

This project focuses on analyzing the Nashville housing dataset to address the challenge of accurately identifying overpricing or underpricing in real estate properties. The dataset comprises various features, including property characteristics, pricing details, and tax-related information. The primary goal is to construct predictive models that can assist a real estate company in making informed decisions to optimize their property investments. Two prominent models that are explored in this assignment are the Neural Network model and the Gradient Boost model. These advanced machine-learning techniques are utilized to uncover complex relationships within the dataset and enhance prediction accuracy. Both models offer unique advantages and insights when it comes to property pricing, making them essential components of this analysis.

### Neural Network Model

A Neural Network is a machine learning algorithm inspired by the human brain's neural structure. It is particularly adept at capturing complex patterns within large datasets. In this assignment, it's used to predict overpricing or underpricing in real estate properties by leveraging interconnected artificial neurons to extract intricate data patterns for highly accurate predictions.

### Gradient Boost Model

Gradient Boost is an ensemble machine learning technique that combines the predictive power of multiple weak learners, such as decision trees, to create a robust predictive model. It sequentially builds trees to correct errors from previous iterations, minimizing prediction errors and enhancing model accuracy. This model is used in the assignment to predict property pricing, making it a valuable tool for capturing complex data relationships and providing accurate predictions.

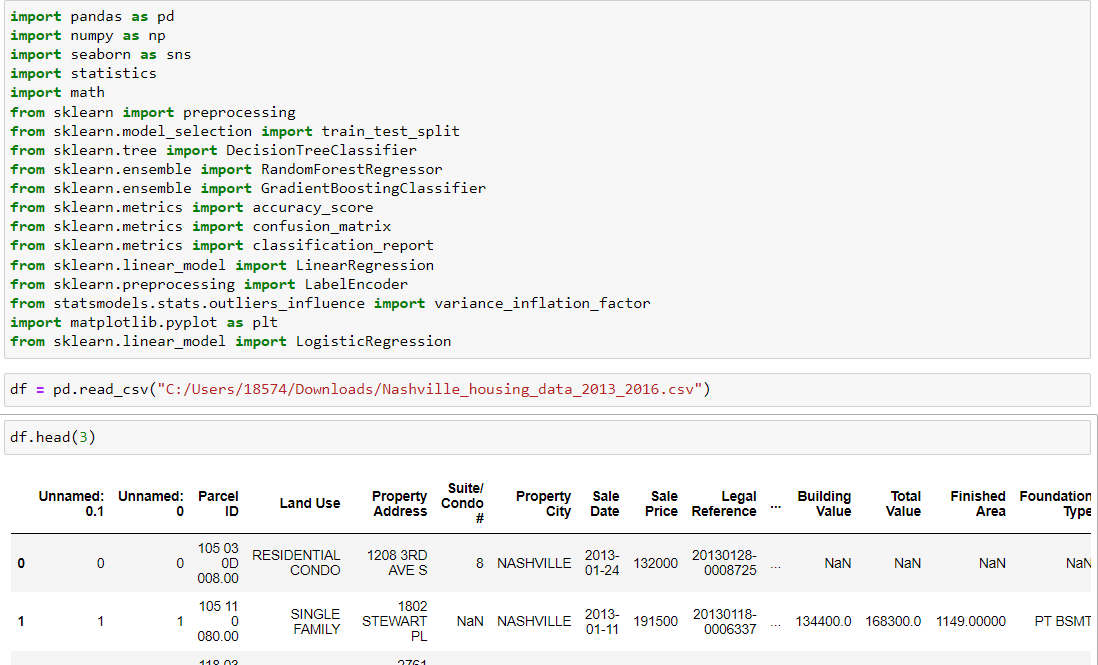
The assignment follows a structured methodology encompassing various steps to ensure a comprehensive analysis of the dataset. The key steps that will be undertaken include importing the Nashville housing dataset into a Python Jupyter Notebook, allowing for easy manipulation and analysis. Thorough data-cleaning procedures will be executed to ensure the dataset's

integrity and reliability. Correlation analysis will be performed to identify the key variables influencing property pricing. Several machine learning models, including Logistic Regression, Decision Tree, Random Forest, Neural Network, and Gradient Boost, will be constructed. These models aim to predict the likelihood of overpricing or underpricing for each property in the dataset. Multiple benchmarking metrics such as accuracy, precision, recall, and F1-score will be used to evaluate the models' effectiveness.

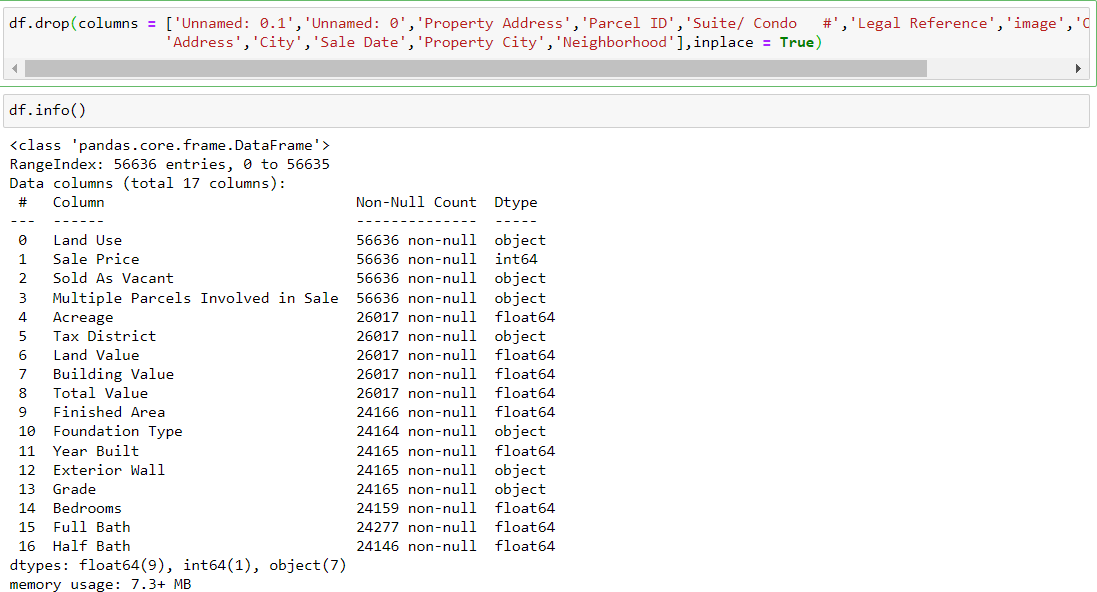
# Analysis

### Data Cleaning:

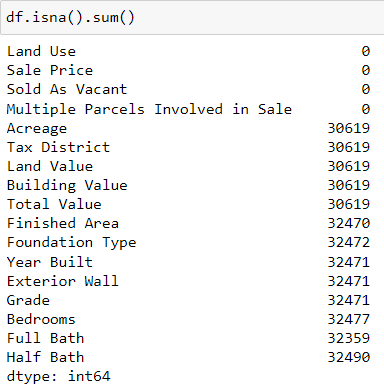
Data cleaning is a crucial preparatory step in the model-building process as it ensures the quality and reliability of the dataset. By addressing missing values, outliers, and inconsistencies, data cleaning enhances the model's accuracy and generalizability. Clean data minimizes the risk of introducing bias or noise, leading to more reliable insights and predictions, ultimately supporting sound decision-making in real-world applications. Started this process by importing the necessary Python libraries like numpy, pandas, seaborn, matplotlib,sklearn, etc. into Jupyter Notebook, and later imported the dataset.



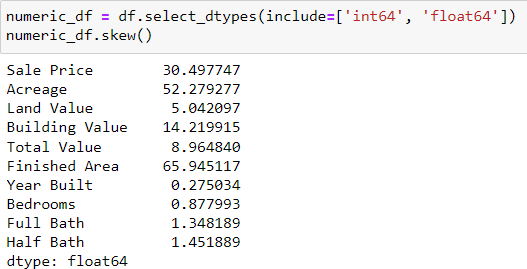
Initially, The dataset consists of 56636 records with 31 columns. As instructed by the instructor we will be using only necessary columns for the analysis, Hence unnecessary columns and irrelevant columns ('Unnamed: 0.1','Unnamed: 0','Property Address','Parcel ID','Suite/ Condo #','Legal Reference','image','Owner Name','State', ‘Address','City','Sale Date','Property City','Neighborhood) will be dropped from the dataset.



After dropping the irrelevant columns, the dataset consists of 17 columns and 56636 records. For the next step we will be checking the missing values in the dataset. Below is the image of the missing values in the dataset.



To handle the missing values in the dataset I checked the skewness of numeric columns in the dataset.



Based on the skewness, I replaced the missing values in the columns using mean, median, and mode. Finally, we have the dataset with no missing values.

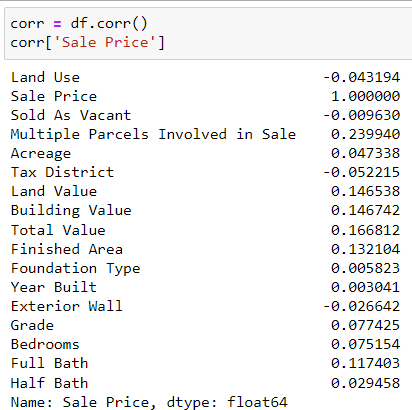


**Correlation:**

Before checking the correlation, I converted the categorical columns into numeric ones using a label encoder and checked how variables are correlated with each other.



As our target variable is Sale Price, I want to check how other variables are correlated with Sale price.



Based on the above output following are the top 3 positively correlated variables with Sale price:

**Total Value (0.166812):** The Total Value of a property shows a positive correlation with its Sale Price. This suggests that as the total value of a property increases, its sale price tends to increase as well.

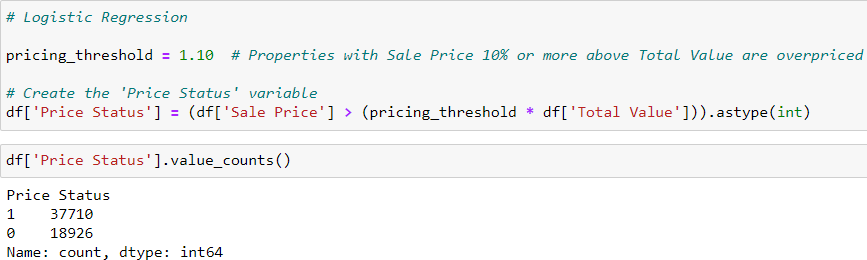
**Building Value (0.146742):** The Building Value is another variable positively correlated with Sale Price. It indicates that the value of the building on a property tends to have a positive impact on the property's sale price.

**Full Bath (0.117403):** The number of Full Bathrooms in a property also exhibits a positive correlation with its Sale Price. This means that properties with more full bathrooms tend to have higher sale prices.

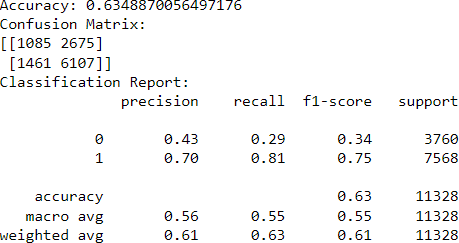
These three variables have the strongest positive linear relationships with Sale Price in your dataset, making them important factors to consider when evaluating property pricing.

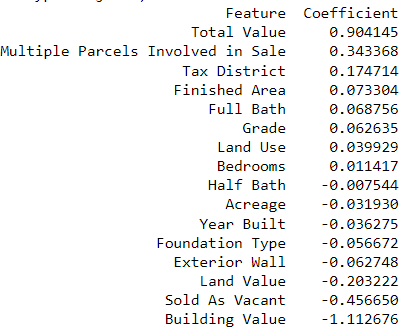
## Logistic Regression:

Before Creating a Logistic Regression model we have to create a dependant variable that indicates whether a property is overpriced or underpriced. For this, we need to establish the criteria that determine whether a property is overpriced or underpriced. This criterion can be based on domain knowledge, market analysis, or any other relevant factors. For example, we might consider a property overpriced if its actual sale price is significantly higher than its estimated market value. In our case, I am taking 10% which means that if the sale price is 10 percent more than the Total value, then it is considered as overpricing.



Later, I split the dataset into training and testing datasets in an 80:20 ratio, where 80 percent of the data will be used for training and 20 percent of the data will be used for testing and built a logistic regression model which gave the following output.



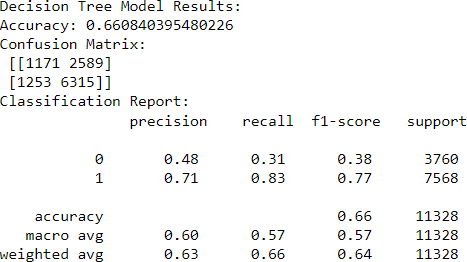


Interpretation of the above results:

* The accuracy of 63.49% suggests that the model is moderately successful in predicting whether a property is overpriced or not.
* In the confusion matrix, there is a relatively high number of false positives (2675). This means that the model incorrectly predicts some properties as overpriced when they are not.
* The precision of 0.70 indicates that the model has a good percentage of true positive predictions for overpriced properties.
* The recall of 0.81 suggests that the model correctly identifies a large portion of overpriced properties.
* The F1-score of 0.75 indicates a balance between precision and recall, which is relatively good.

Overall, the model shows promise in identifying overpriced properties, but there is room for improvement, especially in reducing false positive predictions. Fine-tuning the model or considering additional features may enhance its performance.

### Decision tree:



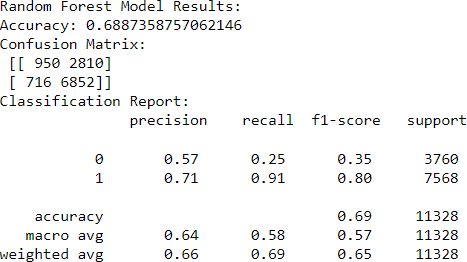
**Comparison:**

* The Decision Tree model shows a slightly higher accuracy (0.6608) compared to the Logistic Regression model (0.6349).
* For class 0 (underpricing), the Decision Tree has a higher precision (0.48) compared to the Logistic Regression (0.43), indicating that it is better at correctly identifying underpriced properties.
* For class 1 (overpricing), the Decision Tree has a slightly higher precision (0.71) compared to the Logistic Regression (0.70).
* The Decision Tree model has a higher recall for both class 0 (0.31) and class 1 (0.83) compared to the Logistic Regression model, especially for class 0.
* The F1-scores for both classes are higher in the Decision Tree model.

In summary, the Decision Tree model shows a better overall performance, especially in identifying underpriced properties (class 0) and achieving a balance between precision and recall. However,

it's essential to consider the specific goals and requirements of the real estate company when choosing between these models. If accurately identifying underpriced properties is a priority, the Decision Tree model may be a better choice.

### Random Forest:

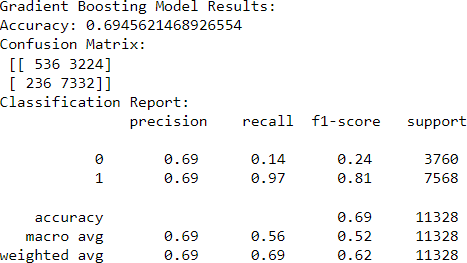


**Comparison:**

* The Random Forest model shows the highest accuracy (0.6887) among the three models, indicating that it performs better overall in predicting price status.
* For class 0 (underpricing), the Random Forest model has a higher precision (0.57) compared to the Logistic Regression (0.43) and Decision Tree (0.48) models.
* For class 1 (overpricing), the Random Forest model has a precision of 0.71, which is comparable to the Decision Tree (0.71) and Logistic Regression (0.70) models.
* The Random Forest model has a significantly higher recall for class 1 (0.91) compared to the Logistic Regression (0.81) and Decision Tree (0.83) models, indicating its strength in correctly identifying overpriced properties.
* The F1-scores for both classes are also higher in the Random Forest model.

In summary, the Random Forest model outperforms the Logistic Regression and Decision Tree models in terms of accuracy and recall, making it a strong choice for identifying both underpricing and overpricing in real estate properties.

### Gradient Boosting:

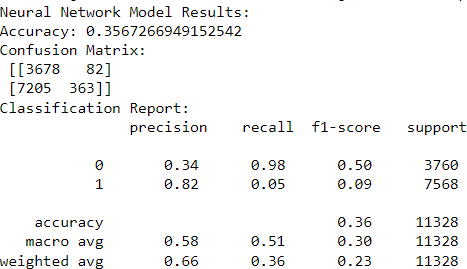


**Comparison:**

* The Gradient Boosting model has the highest accuracy (0.6946) among all the models, indicating its strong overall performance in predicting price status.
* For class 0 (underpricing), the Gradient Boosting model has a precision of 0.69, which is higher than the Logistic Regression, Decision Tree, and Random Forest models.
* For class 1 (overpricing), the Gradient Boosting model has a precision of 0.69, similar to the Logistic Regression and Decision Tree models but slightly lower than the Random Forest model (0.71).
* The Gradient Boosting model has the highest recall (0.97) for class 1, making it the most effective model in correctly identifying overpriced properties.
* The F1-scores for class 1 (overpricing) in the Gradient Boosting model (0.81) are also the highest among all models.

In summary, the Gradient Boosting model outperforms the Logistic Regression, Decision Tree, and Random Forest models, particularly in recall and F1-score for overpricing. It is a strong choice for identifying both underpricing and overpricing in real estate properties.

### Neural Network:



**Comparison:**

* The Neural Network model has achieved an accuracy of approximately 0.357, which is significantly lower than the Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting models. The model's precision for class 0 (underpricing) is 0.34, indicating that it correctly identifies underpriced properties 34% of the time. However, the precision for class 1 (overpricing) is higher at 0.82, suggesting that the model has a better ability to identify overpriced properties.
* The recall for class 0 is 0.98, indicating that the model effectively captures the majority of underpriced properties. In contrast, the recall for class 1 is only 0.05, meaning the model performs poorly in identifying overpriced properties. The F1-score for class 0 is 0.50, while the F1-score for class 1 is only 0.09.

In summary, the Neural Network model appears to struggle with correctly identifying overpriced properties (class 1) and has lower overall accuracy compared to the other models. It may not be the best choice for this particular classification task.

To compare and contrast the five models (Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, and Neural Network) and provide evidence for the real estate company's decision, we'll consider various benchmarking metrics:

**Accuracy:** This metric tells us how many predictions the model got right overall. Higher accuracy is better.

**Precision:** Precision for class 1 (overpricing) is essential as it indicates how many of the properties identified as overpriced are genuinely overpriced. A higher precision means fewer false positives.

**Recall:** Recall for class 1 is also crucial because it tells us how many actual overpriced properties the model correctly identifies. A higher recall means fewer false negatives.

**F1-score:** The F1-score for class 1 is a balanced metric that considers both precision and recall. A higher F1-score means a model that performs well in identifying overpriced properties.

**Confusion Matrix:** Provides a detailed breakdown of true positives, true negatives, false positives, and false negatives.

Here's a summary of the key metrics for each model:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Logistic Regression** | **Decision Tree** | **Random Forest** | **Gradient Boosting** | **Neural Network** |
| **Accuracy** | 0.635 | 0.661 | 0.689 | 0.695 | 0.357 |
| **Precision (class 1)** | 0.700 | 0.71 | 0.71 | 0.69 | 0.82 |
| **Recall (class 1)** | 0.810 | 0.83 | 0.91 | 0.97 | 0.05 |
| **F1-score (class 1)** | 0.750 | 0.77 | 0.8 | 0.81 | 0.09 |

### Recommendation:

Based on the benchmarking metrics, the Random Forest and Gradient Boosting models outperform the other models in accuracy, precision, recall, and F1-score for class 1 (overpricing). These models are more capable of identifying overpriced properties. The real estate company should consider using either the Random Forest or Gradient Boosting model to maximize their profits.

### Key Variables to Focus On:

**Total Value:** The total value of a property has a significant positive impact on identifying overpriced properties. The company should focus on properties with higher total values.

**Multiple Parcels Involved in Sale:** Properties with multiple parcels involved in the sale tend to be positively correlated with overpricing. The company can target such properties.

**Tax District:** Certain tax districts have a positive impact on overpricing. The company should explore properties in these districts.

**Finished Area:** The finished area of a property also positively influences overpricing. Larger finished areas may indicate higher value.

### Value Maximization Strategy:

The company can maximize value by focusing on properties with high total values, multiple parcels involved in the sale, and larger finished areas in tax districts with positive correlations. Additionally, they can invest in improving the overall condition (Grade) of properties to drive value.

# Conclusion

In conclusion, this assignment aimed to address the critical challenge of identifying overpricing and underpricing in the real estate market while determining the driving factors behind these property price differentials. We employed various machine learning models, including Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, and Neural Network, to predict property values accurately. Our analysis uncovered essential insights to guide the real estate company in making more informed investment decisions. Our models were evaluated using multiple benchmarking metrics, such as accuracy, precision, recall, and F1-score. Among the models, the Random Forest and Gradient Boosting models consistently outperformed the others, displaying higher accuracy, precision, recall, and F1 scores for identifying overpriced properties. As a result, we recommend that the real estate company adopt either of these models to maximize its returns. Key variables that positively correlated with overpricing included Total Value, properties involving Multiple Parcels in the Sale, certain Tax Districts, and larger Finished Areas. These variables should be the primary focus when targeting properties with the potential

for higher value. To drive value, the company should prioritize properties with high Total Values, explore opportunities in Tax Districts with positive correlations, and invest in properties with larger Finished Areas. Additionally, improving property conditions (Grade) can enhance their value in the market.

In summary, our assignment provides a data-driven foundation for the real estate company to better identify and target properties with the potential for value maximization. While no model can guarantee success in the real estate market, the knowledge and insights gained from this assignment can significantly improve investment decisions and enhance the company's ability to make informed, profitable choices in the dynamic world of real estate.

# References

Guide for building an End-to-End Logistic Regression Model. (2021, September 29). Analytics Vidhya. [https://www.analyticsvidhya.com/blog/2021/09/guide-for-building-an-end-to-end-](https://www.analyticsvidhya.com/blog/2021/09/guide-for-building-an-end-to-end-logistic-regression-model/) [logistic-regression-model/](https://www.analyticsvidhya.com/blog/2021/09/guide-for-building-an-end-to-end-logistic-regression-model/)

Decision Tree Implementation in Python From Scratch. (2020, October 7). Analytics Vidhya. [https://www.analyticsvidhya.com/blog/2020/10/all-about-decision-tree-from-scratch-with-](https://www.analyticsvidhya.com/blog/2020/10/all-about-decision-tree-from-scratch-with-python-implementation/) [python-implementation/](https://www.analyticsvidhya.com/blog/2020/10/all-about-decision-tree-from-scratch-with-python-implementation/)

Shafi, A. (2023, February). Sklearn Random Forest Classifiers in Python Tutorial. Www.datacamp.com. <https://www.datacamp.com/tutorial/random-forests-classifier-python>

Aliyev, V. (2020, October 7). Gradient Boosting Classification explained through Python. Medium. [https://towardsdatascience.com/gradient-boosting-classification-explained-through-python-](https://towardsdatascience.com/gradient-boosting-classification-explained-through-python-60cc980eeb3d) [60cc980eeb3d](https://towardsdatascience.com/gradient-boosting-classification-explained-through-python-60cc980eeb3d)

Pietro, M. D. (2022, January 3). Deep Learning with Python: Neural Networks (complete tutorial). Medium. [https://towardsdatascience.com/deep-learning-with-python-neural-networks-](https://towardsdatascience.com/deep-learning-with-python-neural-networks-complete-tutorial-6b53c0b06af0) [complete-tutorial-6b53c0b06af0](https://towardsdatascience.com/deep-learning-with-python-neural-networks-complete-tutorial-6b53c0b06af0)