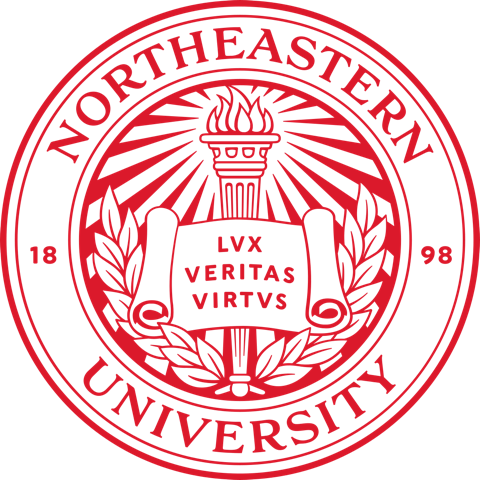
**Module 4 Project**

**Investing in Nashville**

Sunil Raj Thota



College of Professional Studies, Northeastern University ALY 6020 – Predictive Analytics

Prof. Justin Grosz

December 04, 2021

**Introduction:**

This is data on home values in the growing Nashville market. I've recently started working for a real estate firm that plans to make a significant investment in Nashville's expanding economy. They've gathered information on recent sales and want me to construct a guide to help them in correctly locating the greatest value deals when they visit next week. There is concern that homes will sell for more than their asking price, and this information can assist us in observing this. To figure out if it's an over/under price, I'll have to add an outcome variable.

**Part 1:**

To begin, we must cleanse the data and identify any outliers (if any) to ensure that we have high-quality data for the model. We will develop a logistic regression, decision tree, random forest, and gradient boost classifier models to estimate whether the prices are under/ over responses based on the numerous features in the dataset using optimization approaches to see if we can improve the model's accuracy and compare the models, as stated in the question. After evaluating the findings, we will advise the company on which factors contribute to an increase in best value deals. We've imported all of the packages and libraries we'll need for our initial data exploration. Using the scikit-learn to package, you can create, evaluate, and tune various classification models.

**Data Quality:**

To extract certain information from the dataset, the '?' sign is substituted by "NaN" during data filtering. Since the symbol has been changed with NaN, which stands for a null value, these can now be easily identified and computed to verify the sum of the missing values in the collection. There are 56636 records and 32 columns in the dataset. To get started, we've imported all of the packages that are required for undertaking model analysis. Pandas, NumPy, matplotlib, seaborn, train test split, Logistic Regression, Decision Tree Classifier, Random Forest Classifier, Gradient Boost Classifier, and preprocessing are all examples of Python libraries.

**Data Cleansing, Preprocessing, and Exploratory Data Analysis:**

We tested for missing values in the dataset after putting it into a Python environment for further analysis. Missing values must be taken into account because they may have an impact on our analysis and AI models.

Let's see if we can convert the column using astype (). The term "outlier" refers to a data point that differs dramatically from the rest of the dataset. Anomaly in the distance between the values, to be precise. This can happen as a result of experimental errors or measurement variability. In the "Suite/ Condo #" column, there are 50527 null values that can be detected. However, because missing values are sometimes denoted as "Unknown" for categorical data or -1 for numerical data, we must still check for other missing values. There are missing values in almost more than half of the dataset columns like Half Bath, Bedrooms, Foundation Type, Grade, Exterior Wall, Year Built, Finished Area, Full Bath, Owner Name, image, Land Value, Neighborhood, Tax District, Acreage, Building Value, City, Address, Total Vale, State, Property City, and Property Address. And there is no significance in keeping out the Unamed:0, Parcel ID, and Cloumn1 attributes. We investigate the distribution of Sale price to the Total value more fully because several classification algorithms rely on a logit relationship between features and target. To find patterns, we'll look for feature correlations first. I also dropped the rows with empty information which is of no use in our analysis and dropped them accordingly. The new dataset consists of 24013 rows with 19 attributes. I have label encoded the categorical variables and transformed them using the fit\_transform() method

**Part 2:**

**Logistic Regression Model:**

We will be using the Logit function to fit the model with the necessary variables to see the p values and other statistics of each column. I also went ahead and visualized the correlation between the variables using a corr heatmap. Used liblinear as the solver in the Logistic Regression method observed the model fit. And then I have calculated the Accuracy, Precision, Recall, and F1-Scores. The ratio of correctly classified subjects to the total number of subjects is known as accuracy. The ratio of accurately +ve labeled to all +ve labeled is known as precision. A recall is the proportion of those who are in reality to those who are appropriately +ve classified. Precision and recall are both taken into account while calculating the F1 Score. It's the precision and recall's average.

The top 3 significant variables are Sold As Vacant, Tax District, and City is mainly driving the overpricing. From this, we can understand that there is more impact by the city where the houses are located. As the state is divided into various districts, the price is also varied to the district's tax. It also mostly depends on whether the property is sold as vacant or not. We can more focus on these 3 targets and find and invest in markets that have a lot of room for growth. Provide customized direct access to commercial real estate investment options that are institutionally underwritten and data-driven.

**Part 3:**

**Decision Tree Model:**

Decision Tree Analysis is a general-purpose predictive modeling tool with applications in a variety of fields. In general, decision trees are built using an algorithm that determines multiple ways to segment a data set based on certain conditions. It is one of the most popular and practical supervised learning algorithms. Decision Trees are a supervised non-parametric learning method that may be utilized for both classification and regression applications. The goal is to learn simple decision rules from data attributes to develop a model that predicts the value of a target variable. If-then-else sentences are commonly used as decision rules. The rules become more complex as the tree grows deeper, and the model becomes more accurate.

Sold As Vacant, Building Value, and Land Value are the most significant variables in this model. From this, we can understand that there is more impact by these respective overpricing where a greater number of investments are made because of the value of the lands and buildings. It also mostly depends on whether the property is sold as vacant or not. We need to know more details of the building and land to plan our investments and make a decision.

**Part 4:**

**Random Forests Model:**

Random Forest is an ensemble learning-based supervised machine learning technique. I built two Random Forest Classifier models in this project to predict whether a home is expensive or underpriced. With the number of decision trees in the model, the expected accuracy rises. I presented the feature selection process by utilizing the Random Forest model to locate only the most significant features, then rebuilding the model with these features to evaluate how accurate it is. max depth: The total number of splits for all of the forest's trees. bootstrap: Indicates whether or not bootstrap samples should be used while creating trees. max features: The maximum number of features that will be used in node splitting – this is the key difference between bagging trees and random forest as I previously discussed. In most cases, you want a value less than p, where p is the total number of features in your data set. criterion: This is the metric used to evaluate the decision trees' stopping criteria.

Building Value, Finished Area, and Year Built are the most significant variables in this model. From this, we can understand that there is more impact by these respective overpricing where a greater number of investments are made because of the value of the buildings. It depends on the area finished on a particular house. And it also depends on the year of the house built (new/ old). We need to know more details of the building and year to plan our investments and make a decision.

**Part 5:**

**Gradient Boosting Model:**

Gradient boosting is a type of ensemble machine learning method that can be used to solve classification and regression predictive modeling challenges. Gradient boosting is also known as gradient tree boosting, stochastic gradient boosting (a subset of gradient boosting), and gradient boosting machines, or GBM for short. Decision tree models are used to create ensembles. To repair the prediction mistakes caused by past models, trees are introduced to the ensemble one at a time and fitted. The boosting model is a sort of ensemble machine learning model. Models are fitted using a gradient descent optimization approach and any arbitrary differentiable loss function. Gradient boosting gets its name from the fact that the loss gradient is minimized when the model is fitted, much like a neural network. The algorithm outputs hyperparameters that should, and possibly must be modified for each dataset. Although there are many hyperparameters to modify, the following are likely the most important: The model's number of trees or estimators. The model's pace of learning. For stochastic models, the row and column sampling rates. The maximum depth of the tree. The tree's minimal weight. Alpha and lambda are regularization words.

Sold As Vacant, Land Value, and Building Value are the most significant variables in this model. From this, we can understand that there is more impact by these respective overpricing where a greater number of investments are made because of the value of the lands and buildings. It also mostly depends on whether the property is sold as vacant or not. We need to know more details of the building and land to plan our investments and make a decision.

**Part 6:**

**Comparisons, Findings, and Recommendations:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Speed** | **Accuracy** | **Precision** | **Recall** | **MSE** | **ROC AUC Score** |
| Logistic Regression | 0:00:00.11 | 0.735 | 0.66 | 0.025 | 0.26 | 0.57 |
| Decision Tree Model | 0:00:00.05 | 0.755 | 0.89 | 0.09 | 0.24 | 0.59 |
| Random Forest Model | 0:00:00.75 | 0.728 | 0.49 | 0.26 | 0.27 | 0.99 |
| Gradient Boosting Model | 0:00:03.10 | 0.760 | 0.83 | 0.13 | 0.24 | 0.70 |

In both, models, Sold As Vacant and Building Value are the 2 majorly contributing parameters to the prediction if it is overpriced or underpriced. So, I would recommend this company to focus more on the data whether the property sold as vacant or not, and properly analyze the building value based on the markets. From the above results, I would recommend Gradient Boosting Classifier as the best model to go with because of its optimal Accuracy, Recall, Precision, and MSE Scores.

The real estate company has to find and invest in markets that have a lot of room for growth. Provide customized direct access to commercial real estate investment options that are institutionally underwritten and data-driven. When coming to the significant variables in all the models, Sold As Vacant, Building Value, Land Value, Finished Area, Year Built, Tax District, and City are the more significant variables predicting the prices. As these are working, the company has to change its marketing strategy and implement a detailed data-driven decision-making analysis and recommendations to the users.

**Conclusion:**

For the majority of investors, purchasing or selling real estate is one of the most crucial decisions they will make. Selecting a real estate professional/counselor is still an important step in the process. They are well-informed about crucial market elements such as changes in market conditions, market projections, consumer attitudes, optimal locations, timing, and interest rates that affect your unique market areas.

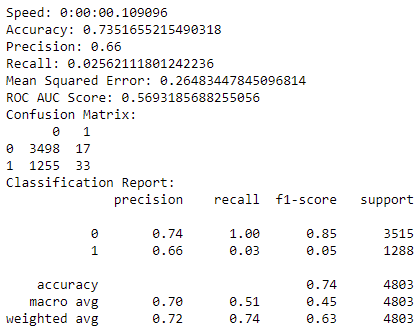
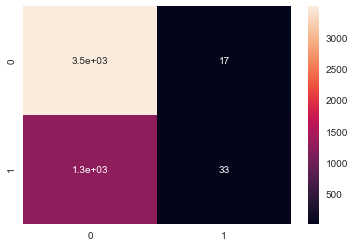
**References:**

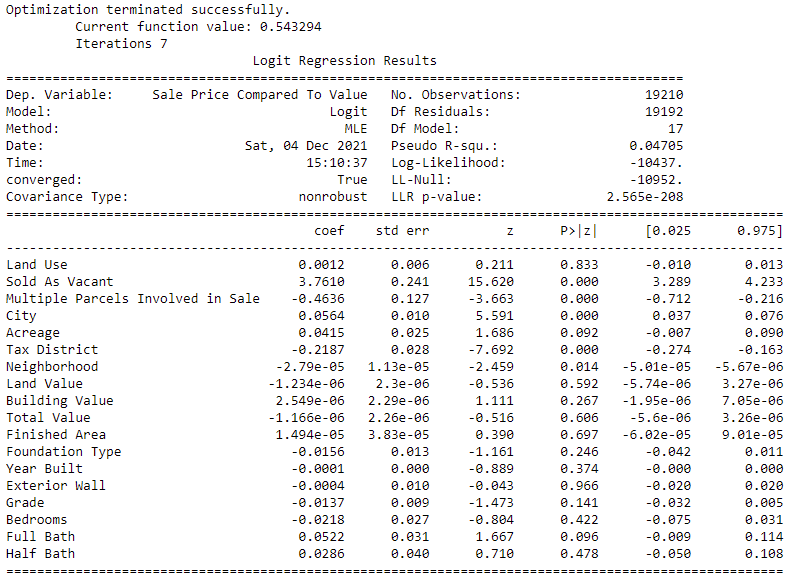
Jason Brownlee. (May 04, 2020). How to Develop a Gradient Boosting Machine Ensemble in Python. *Machine Learning Mastery*.

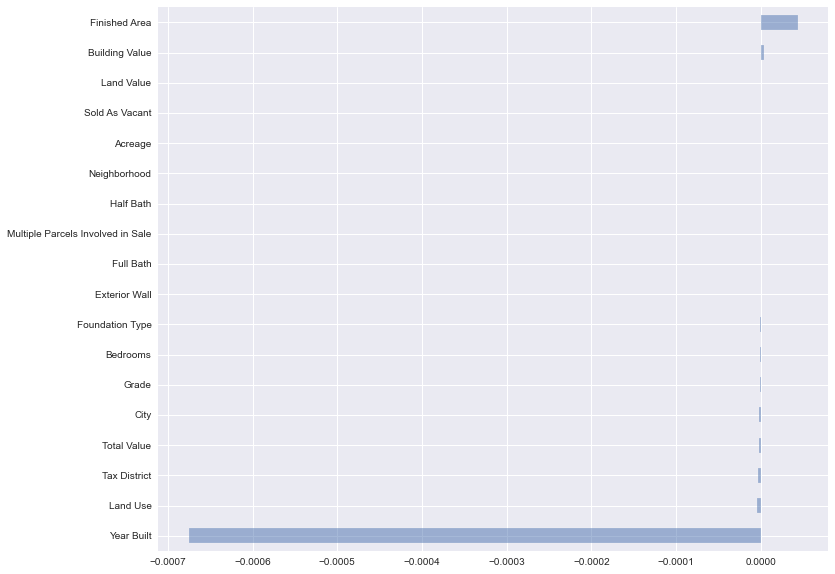
Retrieved from https://machinelearningmastery.com/gradient-boosting-machine-ensemble-in-python/

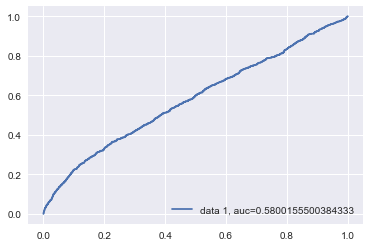
**Appendix:**

**Figure 1: Logistic Regression Results**

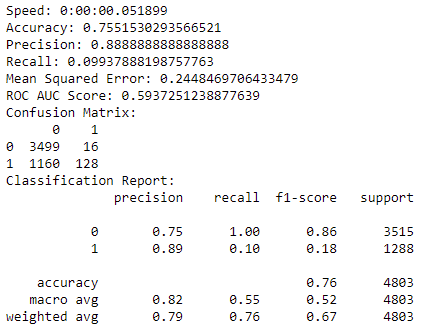
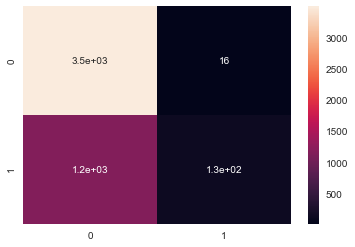
 

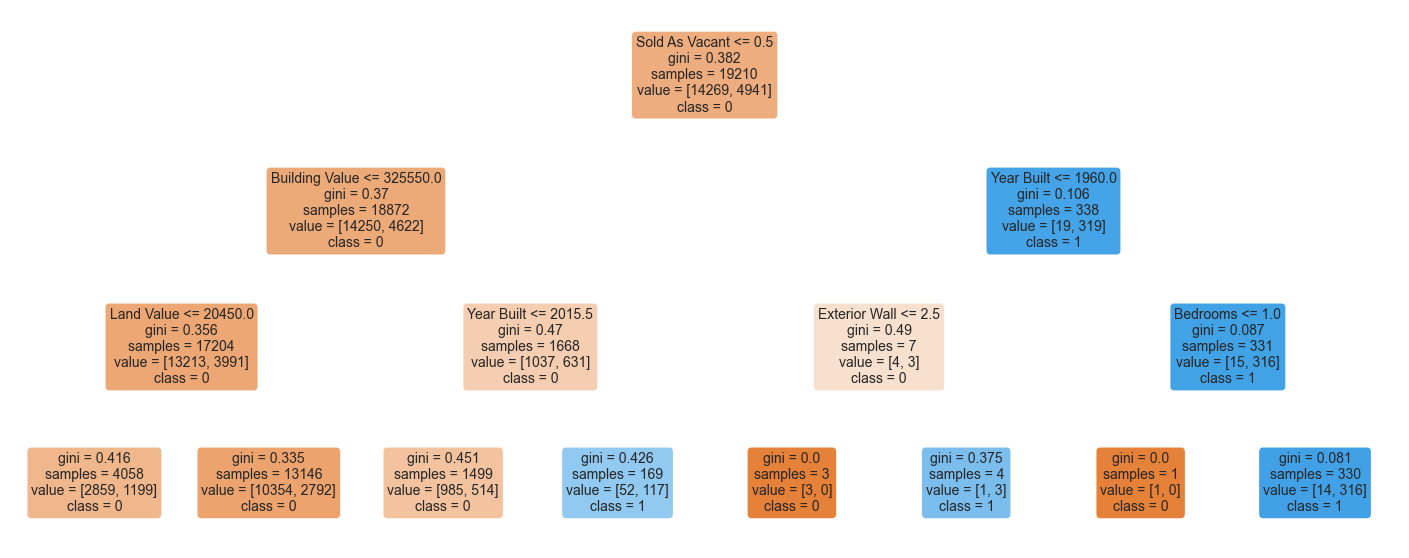


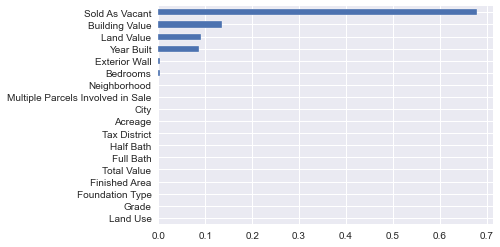




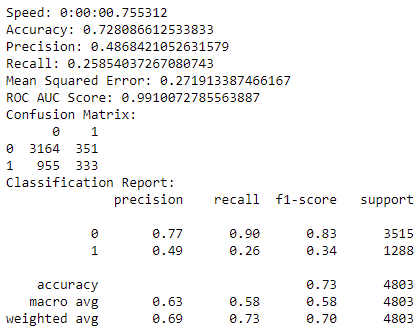
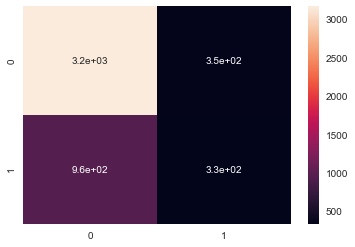
**Figure 2: Decision Tree Classification Results**

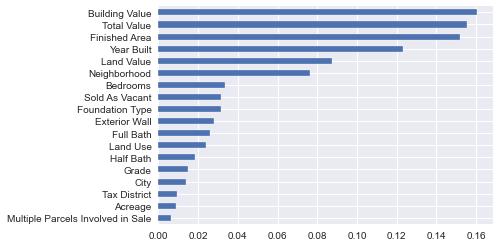
 



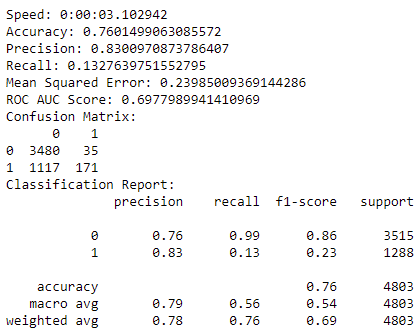


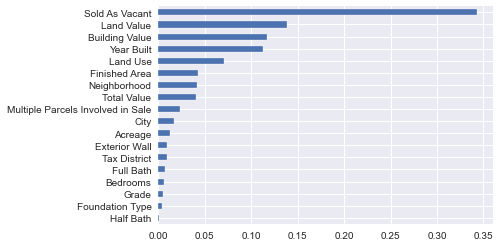
**Figure 3: Random Forest Results**

** **

****

**Figure 4: Gradient Boosting Classification Results**

** **

****