Module 4 Assignment: Investing in Nashville

Dhairyav Jatin Shah

College of Professional Studies, Northeastern University

Professor: Justin Grosz

February 5, 2023

**Introduction**

This dataset contains data about the recent Nashville housing sales. A real estate company is looking to invest in the Nashville area, hence they have acquired this data set including various features of the sale and the property. Using this data set we plant to develop a model to predict if the price of the property is Overpriced or Underpriced. Our target variable is “Sale Price Compared To Value” this will enable us to identify the properties that are being over- or undervalued. The analysis and prediction should help the company understand and get a clear picture of the real estate market in the Nashville area before investing.

**Data cleaning**

There are 22651 records in the dataset, along with 26 variables. The data set has 26 variables total, including 13 categorical date type variables, 11 numerical variables and 2 date type variables. To start the data cleaning process we first check for null values present in the data set. The column “**Unnamed: 0”** is of no use, **“Suite/ Condo #”**  is an empty column ,**“City”** is a duplicate column having the same values as **“Property City”** hence it will be dropped. Other columns containing null values are **“Property Address”, “Property City”, “Finished Area”, “Foundation type”, “Bedrooms” , “Half Bath”, and “Full Bath”.** The columns **“Property Address”, “Property City”** have 2 Nan values hence will drop these 2 columns, **“Foundation type”** being a categorical variable will replace the null value by the mode of the column and other columns we will use the median to replace the mull value as the columns are skewed (fig no 1 in appendix 1) and we don’t want the outliers to shift the mean and give inaccurate value to replace. Further the columns bathrooms, half bath and full bath will be converted into integers from float data type. In the next step we can see that for the categorical variable **Grade** we have one entry each for **SSC, OFC, AAB and OFB** hence we will combine these into one category and name it **MISC.**

**EDA**

Once we have completed the process of data cleaning, we construct a descriptive statistical table to understand the data. From table no 1 (in appendix 2), we can see that the average acreage of the property is 0.45, also we can see that the average building value is $172,220.80 whereas the average land value is $70,133.34, the average finished area of the property is 1915.23, there are 0-11 bedrooms, 0-10 full bathrooms and 0-3 half bathrooms. Further we plot count plots of categorical variables. From fig 2 (in appendix 1) we can see 3 plots, in fig 2.1 we can see that there are almost over 16,000 properties which are over valued whereas less than 6,000 properties are undervalued. In fig 2.2, we can see that over 16,000 properties sold were not vacant were overvalued and over 5000 properties undervalued. The same goes for the graph which shows if multiple parcels were involved in the sales or not. Further in fig 3.1 (in appendix 1), we can see that majority of the are single family houses, having a few properties which include duplexes, residential combos, etc. From fig 3.2 we can see that the foundation type of more than 13,000 properties is Crawl. From fig 4.1 (in appendix 1) we can see that majority properties have brick exterior walls followed by frame and then a combination of brick and frame. In fig 4.2 we can see that more than 15,000 properties lie in the Grade C category followed by Grade B, D, A, X and then remaining MISC. From fig 5.1 and 5.2 ( in appendix 1), we can see that the city with most properties is Nashville and these properties fall under the Urban Services District.

Once we complete the analysis of these categorical variables, we now convert the following categorical variables into discrete numerical variables which will help us build the model easily without creating more columns of Dummy variables, **'Land Use', 'Foundation Type', 'Exterior Wall', 'Grade', 'Property City', 'Tax District', 'Sale Price Compared To Value'.** The dictionary for each value assigned to each category is given in figure number 6 (in appendix 1). Once we have these variables in discrete numerical form we will convert the remaining 2 columns in to the dummy variables and eventually plot a heat map. From fig 7 (in appendix 1), we can see that our target variable **'Sale Price Compared To Value'**  is correlated to which independent variables.

**Analysis**

**Modeling**

Once the initial analysis is complete, we split our data into train and test data frames with a 80:20 ratio. Before we split the data in train and test sets. We divide the data frame into the Y and X data frames having dependent and independent variables respectively. Once we have completely split the data set we fit our first logistic regression model. From the first model we eliminate the insignificant features and refit the model. We can see the insignificant models in table number (in appendix 2). The insignificant features are **'Neighborhood', 'Finished Area', 'Bedrooms', 'Half Bath', 'Full Bath', 'Foundation Type\_new', 'Exterior Wall\_new', 'Property City\_new', 'Acreage'.** From the classification report in table no (in appendix 2), we can see that we fit the model with 75% of prediction accuracy. From table no (in appendix 2) we can see that in our confusion matrix the model is easily able to predict the true positives but is falsely predicting a lot of False positives which is not a good sign for the prediction model. This might also happen because majority of the data we have retrieved has properties listed as overvalued. We now fit the second model i.e Decision Tree Classification. In this model, from table no ( in appendix 2) we can see that the accuracy of prediction is again 75% the same as our logistic regression model. From table no ( in appendix 2), the confusion matrix for Decision Tree Classification we can see similar results as the logistic regression model, just Decision Tree Classification was able to predict more true positives compared to the earlier model, but the false positives are the same which is not good. From fig number 8 ( in appendix 1) we can see the feature importance of the independent variables where if the property was sold vacant or not is considered as the most important variable following with the building value and land value. Figure 9 ( in appendix 1) represents the tree which is formed from the Decision Tree Classifier. Looking at this tree we can also say that majority of the properties are overvalued. The next model we fit is the random forest classifier. From table no ( in appendix 2) the confusion matrix we obtain from the model has predicted 3,377 true positives but it has also predicted more false positives compared to the earlier models. The accuracy of this model is 75%. Figure 10 represents the important features used in this model. The last model we develop is the gradient boosting classifier. The accuracy of this model is 76% also from table no ( in appendix 2) we can see the confusion matrix that we have obtained has predicted the lowest false positives out of all other models.

**Conclusion**

Based on our analysis and modeling, we created 4 classification models. Out of all the 4 created models we can see that gradient boosting classifier has the highest accuracy of 76% and all other models have an accuracy of 75%. We can also see that the confusion matrix of gradient boosting algorithm has 3,359 true positives and 62 true negatives and 1,084 false positives and 25 false negatives. Gradient Boosting algorithm is the algorithm the company could use to predict and understand the market trend before investing as it has predicted the least false positives compared to other models. False positives are comparatively more important than true positives as it predicts the property as overvalued whereas it is not.

**Reference**

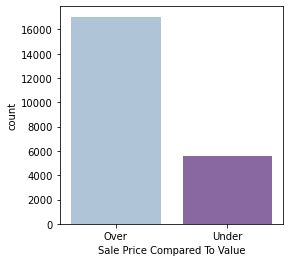
1. Raj.A (Oct 2021) *An Exhaustive Guide to Decision Tree Classification in Python 3.x* <https://towardsdatascience.com/an-exhaustive-guide-to-classification-using-decision-trees-8d472e77223f>
2. N.d. (May 2018) *Understanding Random Forests Classifiers in Python Tutorial* <https://www.datacamp.com/tutorial/random-forests-classifier-python>
3. Aliyev.V (Spet 2020) *Gradient Boosting Classification explained through Python* <https://towardsdatascience.com/gradient-boosting-classification-explained-through-python-60cc980eeb3d>
4. Chouinard.J (May 2022) *How to use Confusion Matrix in Scikit-Learn (with Example)* <https://www.jcchouinard.com/confusion-matrix-in-scikit-learn/>
5. Chouinard.J (May 2022) *How to use Classification Report in Scikit-learn (Python)* <https://datatofish.com/statsmodels-linear-regression/>

**Appendix 1**

**Fig no 1: Distribution Plot**

**A picture containing shoji, crossword puzzle, shrimp

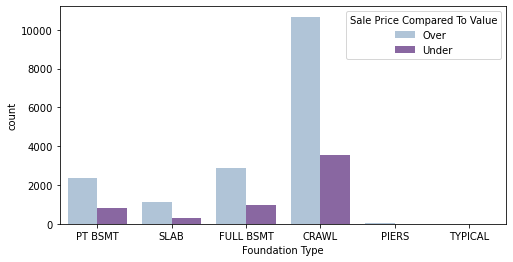
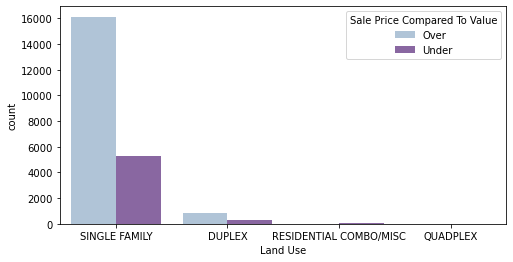
Description automatically generatedFig no 2: Count Plot**



**Chart

Description automatically generated**

**Fig no 3: Count Plot**



**Fig no 4: Count Plot**

**Chart, histogram

Description automatically generated**

**Chart

Description automatically generated with low confidence**

**A picture containing histogram

Description automatically generatedFig no 5: Count Plot**

**Chart, histogram

Description automatically generatedFig no 6: Heat Map**

**Chart, treemap chart

Description automatically generated**

**Fig no 7: Feature Importance of Decision Tree Classification**

**Chart, bar chart

Description automatically generated**

**Fig no 9: Feature Importance of Random Forest**

**Chart, bar chart

Description automatically generated**

**Fig no 10: Feature Importance of Gradient Boosting**

**Chart, bar chart

Description automatically generated**

**Fig no 8 : Tree of Decision Tree Classification**

**A picture containing timeline

Description automatically generated**

**Appendix 2**

**Table no 1: Descriptive Statistics**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **count** | **mean** | **min** | **max** |
| **Acreage** | 22649.0 | 0.454724 | 0.04 | 17.5 |
| **Neighborhood** | 22649.0 | 4432.803832 | 107.00 | 9530 |
| **Land Value** | 22649.0 | 70133.345225 | 900.00 | 1869000 |
| **Building Value** | 22649.0 | 172220.807762 | 1400.00 | 5824300 |
| **Finished Area** | 22649.0 | 1915.232650 | 450.00 | 19728.25 |
| **Year Built** | 22649.0 | 1961.942911 | 1832.00 | 2017 |
| **Bedrooms** | 22649.0 | 3.104773 | 0.00 | 11 |
| **Full Bath** | 22649.0 | 1.887191 | 0.00 | 10 |
| **Half Bath** | 22649.0 | 0.268886 | 0.00 | 3 |

**Table 2: Logistic Regression Model**

**Table

Description automatically generatedTable 2.1 : Logistic Regression Model Results**

**Table 2.2 : Confusion Matrix of Logistic Regression Model**

|  |  |  |
| --- | --- | --- |
|  | **Yes** | **No** |
| **Yes** | 3367 | 17 |
| **No** | 1106 | 40 |

**Table 2.3 : Classification Report of Logistic Regression Model**

Table

Description automatically generated with medium confidence

**Table 3 : Decision Tree Classification, Random Forest and Gradient Boosting**

**Table 3.2 : Confusion Matrix for all 3 models.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Decision Tree** | |  | **Random Forest** | |  | **Gradient Boosting** | |
|  | **Yes** | **No** |  | **Yes** | **No** |  | **Yes** | **No** |
| **Yes** | 3375 | 9 | **Yes** | 3377 | 7 | **Yes** | **3359** | **25** |
| **No** | 1106 | 40 | **No** | 1118 | 28 | **No** | **1084** | **62** |

**Table 3.3 : Classification report of all 3 models.**

**A picture containing table

Description automatically generated Decision Tree**

**Table

Description automatically generated with low confidence**  **Random Forest**

**Table, calendar

Description automatically generated Gradient Boosting**