**Module 4: Investing in Nashville**

ALY6020 – Predictive Analytics, Northeastern University, Boston

Professor Justin Grosz

CRN: 20387

03/12/2023

Submitted by: Abhinav Jain

**Introduction**

This report's objective was to build a model that would assist the business in finding the greatest value transactions by analyzing the real estate sales data of the expanding Nashville area. The dataset includes details on recent real estate transactions, such as the sale price and the property's valuation. The variable "Sale Price Compared to Value" will be the focus of our investigation to identify whether properties are being over or undervalued.

After analysis, data cleaning, and exploration of the dataset. We eliminated any erroneous or incomplete data points to guarantee the correctness of our model. Afterward, concentrated on the prediction variable "Sale Price Compared to Value" to spot any patterns in the data.

According to our data, certain properties were inflated while others were being undervalued. We attempted to create a machine-learning model utilizing logistics, a decision tree, a random forest, and a gradient boost that examined the historical data to discover patterns and trends in order to determine the greatest value transactions. The program was taught to forecast a property's sale price based on its worth, location, and other important elements.

**Data Cleaning**

To verify the accuracy and dependability of our model, we meticulously scrutinized each data point in our research of the real estate sales data. While the analysis found that some of the data points, including Property Address (2), Finished Area (1), Foundation Type (1), Bedrooms (3), Full Bath (1), and Half Bath (108), were null information for certain variables. Decide to remove the missing values from the dataset to make sure that our analysis was supported by correct and full data. With the use of this strategy, we were able to concentrate on the data points that had comprehensive information for all pertinent variables, ensuring that the foundation of our model was solid and trustworthy.

To strengthen our analysis and improve the precision of our predictions by eliminating the missing variables. To conduct analysis, we are convinced that the prediction model will offer the business useful insights into the real estate market in the expanding Nashville region and that it will assist the business in identifying the greatest value offers for their investment.

**Exploratory Data Analysis (EDA)**

**A picture containing chart

Description automatically generated**The dataset was presented and includes the sale prices of the properties, 22,536 records, and 17 columns. Data analysis revealed that one property, with a selling price of 16,979, was undervalued, while another, with a sale price of 5,557, was overpriced.

By transforming the columns into dummy variables, a total of 48 columns were produced, allowing for further analysis of the data. The accuracy of future selling prices was then predicted using the model. To uncover the connections among the variables in the dataset, a correlation matrix was also made. The research showed a strong association between the finished area and building value (correlation coefficient of 0.86), but only a weak correlation (correlation coefficient of -0.011) between land value and the year of construction.  The research involves a review of the relationships between various variables and shows the undervalued and overpriced qualities in the dataset. The logistic regression model offers a method for forecasting future selling prices and may be applied to guide future real estate market decisions.

**Logistics Regression**

While implementation of the logistics regression model achieved a total accuracy of 0.75891 using the logistic regression model that was trained on the provided dataset. This indicates that 75.69% of the model's predictions were accurate. The model's accuracy was 0.759, which means that when it correctly predicts a favorable outcome, it does so 75.94% of the time. The model's recall was 0.998, which indicates that it accurately predicted 99.78% of the favorable outcomes.

From the confusion matrix, we can see that the model had many false positives (1619) compared to false negatives (11). This means that the model tended to predict positive outcomes more often than was the case. In this logistic regression model achieved a good level of accuracy and recall but had a relatively low precision due to the high number of false positives. Further investigation may be required to understand the reasons for this and to identify ways to improve the model's performance.

**Decision Tree:**

By implementing the decision tree model total accuracy of 0.7649 was based on the decision tree model trained on the provided dataset. This indicates that 76.49% of the model's predictions were accurate. The model's accuracy was 0.77, which means that when it predicts a favorable outcome, it was accurate 77% of the time. For the negative class, the model's recall was 0.04, while for the positive class, it was 1.00. This suggests that while only a very tiny number of the bad events were accurately predicted by the model, all the positive ones were. The confusion matrix reveals that the model produced many more false positives (1566) than false negatives (23). This indicates that the model was more likely to forecast favorable results than they really were. The decision tree model exhibited a poor recall for the negative class but excellent levels of accuracy and precision. This suggests that while just a tiny part of the bad events was accurately predicted by the model, all of the positive ones were. To comprehend the causes of this and to find solutions to enhance the model's performance, particularly for the negative class, more research may be necessary.

**Random Forest:**

After implementation of the Random Forest Model achieved an overall accuracy of 0.7615 using the Random forest model that was trained on the provided dataset. This indicates that 76.15% of the model's predictions were accurate. The model's accuracy was 0.76, which means that when it predicts a favorable outcome, it was accurate 76% of the time. The model's recall for the negative class was 0.02 whereas it was 1.00 for the positive class. This suggests that while only a very tiny number of the bad events were accurately predicted by the model, all the positive ones were.  This was observed from the confusion matrix that the model had more false positives (1606) than false negatives (6). This indicates that the model was more likely to forecast favorable results than they really were. In fact, the random forest model exhibited poor recall for the negative class but excellent levels of accuracy and precision. This suggests that while only a very tiny number of the bad events were accurately predicted by the model, all the positive ones were. To comprehend the causes of this and to find solutions to enhance the model's performance, particularly for the negative class, more research may be necessary.

**Gradient Boost Model:**

The gradient boost model trained on the given dataset; we achieved an overall accuracy of 0.7633. This means that 76.33% of the predictions made by the model were correct. The precision of the model was 0.77, which indicates that when the model predicts a positive outcome, it was correct 77% of the time. The recall of the model was 0.06 for the negative class and 0.99 for the positive class. This means that the model correctly identified 99% of the positive outcomes, but only a small proportion of the negative outcomes. From the confusion matrix, we can see that the model had many false positives (1639) compared to false negatives (54). This means that the model tended to predict positive outcomes more often than was the case.

In conclusion, the gradient boost model achieved a good level of accuracy and precision but had a relatively low recall for the negative class. This means that the model correctly identified 99% of the positive outcomes, but only a small proportion of the negative outcomes. Further investigation may be required to understand the reasons for this and to identify ways to improve the model's performance, especially for the negative class.

|  |  |  |  |
| --- | --- | --- | --- |
| Prediction Matrix | Accuracy (%) | Precision (%) | Recall (%) |
| Logistics Regression | 75.69 | 75.94 | 99.78 |
| Decision Tree | 76.49 | 77 | 100 |
| Random Forest | 76.15 | 76 | 100 |
| Gradient Boost | 76.33 | .77 | 99 |

**Conclusion:**

In the given results, the decision tree model achieved the highest accuracy and precision scores, both at 76.49% and 77%, respectively, and achieved 100% recall for both positive and negative classes. Therefore, the decision tree model appears to be the best model among the given models.

However, it's important to note that the performance of the models may vary based on the specific context of the problem, and it may be necessary to consider other factors such as model complexity, interpretability, and computational requirements when selecting the best model for a particular use case.

Advise the business to concentrate on undervalued properties considering our findings. The corporation will be able to locate these assets and strike the greatest agreements with the help of our model. The business may optimize its investment and benefit from the expanding Nashville real estate market by utilizing our methodology to find the greatest value opportunities.

To take advantage of these opportunities, the business should consider renovating these undervalued properties with extra accessories and facilities. This can help increase their overall value and attractiveness to a potential buyer. For example, adding modern amenities such as smart home technology, energy-efficient appliances, and upgraded fixtures can significantly increase the appeal of a property and justify a higher asking price.

**Appendix:**

**Correlation:**

Application, treemap chart

Description automatically generated with medium confidenceA picture containing graphical user interface

Description automatically generated

**Decision Tree**

**Graphical user interface

Description automatically generated with low confidence**