

**Module 4 Project**

**ALY 6020**

Name: Kavitha Palthayvath

Instructor’s name: Justin Grosz

Due Date: 08/08/2021

**Abstract**

Being a part of a company, which is planning to make a huge investment into the growing Nashville area, in this paper I am going to outline some important factors using various machine learning models like logistic regression, decision tree, random forest, and gradient boost model. With the help of these models, we will find the important factors that contribute to predicting if the Sale Price of a property is under/overvalued which will directly contribute to sorting out the best deals in which the company should invest.

**Introduction**

The dataset provided to us has details about various properties in and around Nashville area. To use the given data most usefully we performed a through data analysis and make the dataset ready by performing data cleaning and then went on using the cleaned data to train different models and compare their performance metrics to find the best suitable model.

**Analysis**

**Data Cleaning**

While going over all the variables in dataset, we found the ‘States’ variable of no use and removed it as it mostly had empty values, there were a few empty values in ‘Property Address', ‘Property City', 'Finished Area', 'Foundation Type', 'Bedrooms', 'Full Bath', 'Half Bath'. Since the number of records in which we found empty values were negligible compared to the total number of records, we removed these records. Finally, since ‘Property City' and ‘City’ had the same entries, we removed the ‘Property City' variable. Moreover, we discovered that most of the variables are categorical and only a few of them are numeric. The implication of this being that we need to transform a lot of variables by using OneHotEncoding, which after analysis came out to be ~ 65000 columns, using a dataset with these many columns for fitting models would consume a lot of memory and will be time consuming and is not advisable. We instead chose to use LabelEncoding which can be readily used with Decision Trees, Random, Forests and Gradient Boost as well.

**Model Comparison and Analysis**

***Logistic Regression***

We trained a logistic regression model and used automated feature selection with RFE on that model, we re-trained the model with the suggested variables and received the model score/accuracy of 76.24% with a precision of 54.45% and recall of 0.14 with an execution time of 0.25 seconds.

***Decision Tree***

Next we trained a decision tree model and used automated feature selection with RFE on that model, we re-trained the model with the suggested variables and received the model score/accuracy of 70.6% with a precision of 40.17% and recall of 0.42 with an execution time of 0.23 seconds. Though this model was faster, but the accuracy and precision of this model was lesser as compared to logistic regression. Among Logistic Regression and Decision Tree model, logistic regression considered ‘Tax District’ as an important variable, whereas decision tree model considered ‘Acreage’ as an important variable.

***Random Forest***

After Decision Tree, we trained a random forest model (with 100 estimators) and used automated feature selection with RFE on that model, we re-trained the model with the suggested variables and received the model score/accuracy of 78.92% with a precision of 62.52% and recall of 0.33 with an execution time of 5.11 seconds. This model gave us a good accuracy and precision boost but took exponentially more time to execute as compared to logistic regression and decision tree model. To correctly leverage the power of Random Forests we re-trained the model with the same features but with more number of trees/estimators(n\_estimators=500), this model gave us the model score/accuracy of 79.08% with a precision of 62.62% and recall of 0.34 with an execution time of 25.66 seconds. As intended, increasing the number of trees in decision tress model increased the accuracy and precision but again increased the execution time substantially.

***Gradient Boosting***

Lastly, we trained a gradient boosting classifier model (with 200 estimators) without any feature selection to predict the under/over value of a property, this model gave us the model score/accuracy of 79.25% with a precision of 67.30% and recall of 0.28 with an execution time of 10.00 seconds. This is a pretty good model as compared to random forest; hence we decided to use feature selection and make better prediction by retraining this model. We used gradient boosting’s internal feature for feature selection, as compared to decision tree and random forest, gradient boosting added 'Sold As Vacant', 'Multiple Parcels Involved in Sale', 'Exterior Wall', 'Grade' as important features as well. On retraining the model with these features, this model gave us the model score/accuracy of 79.17% with a precision of 66.80% and recall of 0.28 with an execution time of 10.27 seconds. The feature selected gradient boosting model gave almost the same performance but while considering only important features. To make the feature selected gradient boosting model even better we experimented using different number of trees and got the most optimal model (n\_estimators=65). On retraining the model with these many trees, this model gave us the model score/accuracy of 79.32% with a precision of 70.17% and recall of 0.26 with an execution time of 3.35 seconds. This model is the best model we have trained so far it gave better accuracy, precision, recall, and an exponential decrease in execution time.

**Business Recommendation**

To summarize, as per our analysis the various models we trained performed as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **Execution**  **Time (sec)** |
| Logistic Regression | 76.24% | 54.45% | 0.144 | 0.255 |
| Decision Tree | 70.62% | 40.17% | 0.422 | 0.238 |
| Random Forest with 100 trees | 78.92% | 62.52% | 0.334 | 5.11 |
| Random Forest with 500 trees | 79.08% | 62.62% | 0.348 | 25.66 |
| Gradient Boosting without feature selection | 79.25% | 67.30% | 0.287 | 10.00 |
| Gradient Boosting with feature selection | 79.17% | 66.80% | 0.286 | 10.27 |
| Gradient Boosting with feature selection and optimal number of trees | 79.32% | 70.17% | 0.261 | 3.35 |

On comparing the performance of these models, we can conclude that **Gradient Boosting with feature selection and optimal number of trees (65) is the best performing model** as it has the right balance of accuracy, precision, recall, and execution time as compared to other models. The key factors that can help in predicting the Selling Price Compared to value of a property are, the address, Sale date of the property, is the property Vacant or Sold ? , if there are multiple Parcels involved in Sale ? , Acreage , Neighborhood, Land Value, Building Value, Finished Area, Year Built, Exterior Wall, and the Grade of the property.

**Conclusion**

The ability to accurately predict if the property is under or overvalued when compared to its selling price, enables the real estate company to make right decision to invest or not. We are leveraging this fact and have created a model to do exactly this. Using the insights provided as per our analysis the company will be able to pick the right property in Nashville based on the features provided so that they can make the right investment as per their needs.

**References**

Płoński, P. (2020, June 22). *Visualize a Decision Tree in 4 Ways with Scikit-Learn and Python*. Retrieved from mljar: https://mljar.com/blog/visualize-decision-tree/

Singh, H. (2018, November 3). *Understanding Gradient Boosting Machines*. Retrieved from towards data science: https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab