Gaby Masak

D600 – Statistical Data Mining

Task 2: Logistic Regression Analysis

10/2/2024

**Part 1: GitLab Repository URL**

The GitLab repository for this task can be found following this link:

<https://gitlab.com/wgu-gitlab-environment/student-repos/gmasak/d600-statistical-data-mining/-/blob/D600Task1/D600Task2.py>

The repository history can be found in the attached link, noting that technical difficulties with both building a pipeline and generating a token from the WGU GitLab led to the majority of the coding to be developed before the initial commit could be made.

**Part 2: Purpose of Logistic Regression Analysis**

Data analytics has many practical applications in every industry. For the real estate industry, logistic regression analysis can be applied to housing data to predict or estimate the probability that a specific event will occur. For example, whether a house will be classified under luxury or not. Understanding how certain metrics, also known as the independent variables, aid in the classification is essential to applying analytics to development. Real estate agents can use this information to advise homeowners and builders which features are most likely to increase the value of a property or make it more attractive to potential buyers. By leveraging data-driven insights, agents can provide more accurate and strategic recommendations, ultimately enhancing decision-making processes and optimizing market outcomes. This approach not only benefits individual transactions but also contributes to broader trends and patterns in the real estate market, leading to more informed and efficient industry practices.

**Part 3: Summary of Data Preparation Process**

Five variables (four independent) were selected to predict whether houses were considered luxury or not: renovation quality, garage, fireplace and number of bedrooms.

Table 1: Descriptive Statistics of Variables

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Variable | Type | Count | Mode | Range | Max/Min |
| Luxury | Dependent | 7000 | 1 | 1 | 1/0 |
| Renovation Quality | Independent | 7000 | 10 | 9.99 | 10/0.01 |
| Garage | Independent | 7000 | No (0) | 1 | 1/0 (Yes/No) |
| Fireplace | Independent | 7000 | No (0) | 1 | 1/0 (Yes/No) |
| Number of Bedrooms | Independent | 7000 | 3 | 6 | 7/1 |

Univariate and Bivariate Distributions of the Dependent and Independent Variables:

Luxury

A graph of a home

Description automatically generated with medium confidence

Figure 1: Bar chart of Luxury Home Description

Renovation Quality

A diagram of a box plot

Description automatically generated

Figure 2: Boxplot of Renovation Quality

A graph of a line graph

Description automatically generated

Figure 3: Density Plot of Renovation Quality

A graph of a function

Description automatically generated

Figure 4: Histogram of Renovation Quality

Fireplace

A graph of a number of homes with no fireplaces

Description automatically generated

Figure 5: Bar plot of Fireplace Distribution

Number of Bedrooms

A diagram of a box plot

Description automatically generated

Figure 6: Boxplot of Number of Bedrooms

A graph of a function

Description automatically generated

Figure 7: Density Plot of Number of Bedrooms

A graph of a number of bedrooms

Description automatically generated

Figure 8: Histogram of Number of Bedrooms

Garage

A graph of a number of homes with no garage

Description automatically generated

Figure 9: Bar chart of Distribution of Garage

Luxury vs. Renovation Quality

A graph with blue rectangles and black lines

Description automatically generated

Figure 10: Boxplots of Renovation Quality Distribution Grouped by Luxury Designation

Luxury vs. Fireplace

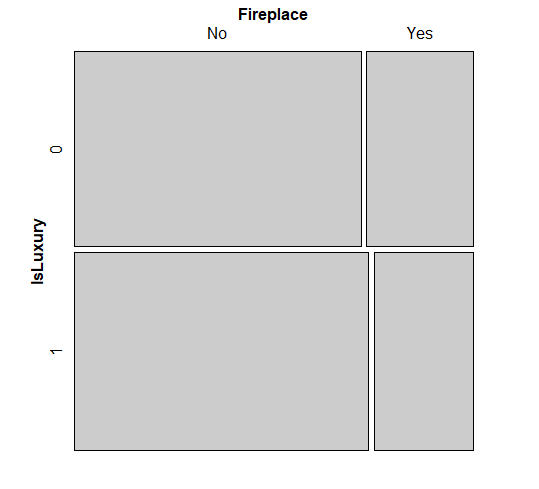


Figure 11: Mosaic Plot of Luxury vs Fireplace

Luxury vs. Number of Beds

A graph with blue rectangles

Description automatically generated

Figure 12: Boxplots of Number of Bedrooms Distribution Grouped by Luxury Designation

Luxury vs. Garage

A grid of gray squares

Description automatically generated

Figure 13: Scatterplot of Price vs. Backyard Space

**Part 4: Analysis and Results**

The 7000 rows were separated randomly into a training set with 5600 rows and a testing set with 1400 rows. These sets can respectively be found in the attached files D600TrainingSet2.csv and D600TestSet2.csv. After separating the data into training and testing sets, the data can be analyzed.

A computer screen shot of a black screen

Description automatically generated

Figure 14: Screenshot of Regression Results

A screenshot of a computer

Description automatically generated

Figure 15: Screenshot Confusion Matrices for Training and Test

Two confusion matrices are displayed in Figure 15 for the training and test sets, representing True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). The designation of true means that the model correctly predicted the result, either positive (luxury) or negative (non-luxury). The designation of false indicates that the results were not correctly predicted. In the matrices above, while there are false positives and negatives, there are also almost double the amount of true values, an observation that is shared by both the training and test sets.

**Part 5: Summary of Data Analysis**

1. Libraries Used in Python and R
   1. Python:
      1. pandas: Used to analyze and manipulate large datasets.
      2. numpy: Used to manipulate arrays and matrices.
      3. statsmodels.api: Used for classes and functions of various statistical methods for statistical tests and data exploration.
      4. sklearn.model\_selection: Used for model selection, analyzing data, training and testing models.
         1. train\_test\_split: Used to randomly split the data frame into training and testing sets, specifying the percentage of test data, and grounding the results.
      5. sklearn.metrics: Used to calculate various prediction errors.
         1. mean\_squared\_error: Used to calculate the mean squared error between real and predicted values.
         2. accuracy\_score: Used for classification models to compute accuracy.
         3. confusion\_matrix: Used with classification algorithms to evaluate for true positives, true negatives, false positives, and false negatives.
   2. R:
      1. readxl: Used to import data from Excel and CSV files.
      2. forcats: Used to handle categorical variables.
      3. dplyr: Used to manipulate data by transforming, filtering, and summarizing.
      4. ggplot2: Used to create visualizations like boxplots, density graphs, etc.
      5. vcd: Used to create mosaic plots.
2. Method Used to Optimize Model and Justification

Similar to linear regression, it is imperative that the model only uses relevant predictor variables for logistic regression. As such, various techniques can be applied to optimize the model, such as forward stepwise selection. Forward stepwise selection focuses on adding singular predictor variables to the model in order of statistical significance, which is an efficient and simple approach that is effective in removing irrelevant variables from the model. When the risk of overfitting is removed, the model can be calculated. Ultimately, only the number of bedrooms and renovation quality are considered, with fireplace and garage excluded.

1. Summary of Assumptions of Logistic Regression

There are several assumptions with logistic regression. The first is that the model results in the prediction of a binary dependent variable. The binary outcome can be either yes or no, 0 or 1, true or false, or another two-option categorical output. The model developed in this task involved luxury or non-luxury predictions, which were defined by 1 (luxury) or (non-luxury). Within the code, this is expressed by the line, “y\_test\_pred\_class = (y\_test\_pred >= 0.5).astype(int)” which forces the outcome of the prediction to be an integer or either 0 or 1, rounding up or down.

A secondary assumption is independence between observations, meaning that the predictions of previous properties will not affect the future categorization of other properties as luxury or non-luxury. An assumption also exists where there is no perfect or high correlation between predictor variables, meaning that the independent variables used to calculate the dependent variable should not overlap or correlate with one another. This can be assessed by visualizing the calculated variables over time. In Figure 16, no pattern is established over time and all recorded visualizations are random meaning that there is independence between observations.

A blue line with a red line

Description automatically generated

Figure 16: Calculated Residuals Over Time

A third assumption lies with the predictor variables, establishing that no extreme multicollinearity should exist between them. This was simple to check, as only two predictor variables were present with a VIF of 1.0943, as seen in Figure 17, indicating that the number of rooms and renovation quality variables lack correlation.

A black background with white text

Description automatically generated

Figure 17: Screenshot of VIF Output

Lastly, the fourth assumption is that no extraneous variables are included. This assumption was met when applying forward stepwise selection, which omits variables garage and fireplace. The code is below in Figure 18.

A screen shot of a computer program

Description automatically generated

Figure 18: Python Code for Forward Stepwise Selection

1. Regression Equation:

The equation formed by logistic regression represents the probability that a house will be defined as luxury. If the equation calculates to be 0.5 or higher, the algorithm will decide that a 50% chance or greater means that the house is luxury. Lower and the algorithm will declare that the house is non-luxury. Compared to linear regression, the coefficients work differently in logistic regression, as the impact of their weight is less direct due to their presence as the exponent of Euler’s number (e). An increase in their value still increases the probability that the house will be defined as luxury, with the number of bedrooms and its coefficient of 0.5762 more heavily affecting this increase compared to renovation quality and its coefficient of 0.3049.

1. Discuss the Model Metrics and Implications of Results:

As observed in Figures 14 and 15, the training and testing accuracies were 61.20% and 62.79% respectively, suggesting that the model generalizes well to unseen data and is not overfitting the training data. The confusion matrices further indicate that the model performs consistently, as the proportions of true and false positives and negatives remained approximately constant between the training and test sets, with approximately double the amount of true values compared to false values. Comparing the Mean Square Errors also aids in this analysis. That said, while the model performs well, there is still room for improvement. The relatively high number of false positives and false negatives indicates that the model could benefit from further tuning or additional features to improve its predictive power. Additional predictor variables could be explored or added.

1. Recommendations

Real estate can be lucrative, but there are also steep financial risks. These risks can be mitigated by using logistic regression to identify properties with high potential for luxury classification. By leveraging predictive models, real estate organizations can make informed decisions about pricing, marketing, and investment strategies. This analysis can also benefit decisions on remodeling and flipping properties, as the presence of a garage was seemingly unimportant to the definition of a luxury property, but the number of bedrooms was, meaning that it would be beneficial for the flipper to convert the garage into additional bedrooms. This data-driven approach helps in targeting the right audience, setting competitive prices, and optimizing resources, ultimately reducing financial risks and enhancing profitability.

**Part 6: Panopto Audiovisual Presentation**

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=78c30688-6995-4ef8-a4c2-b21300267cd3>