**Task 2: Logistic Regression Analysis**

Joseph Emmanuel Cayetano

Western Governors University

D600: Statistical Data Mining Task 2

November 15, 2024

1. **Create your subgroup and project in GitLab**

**Gitlab repository URL:** https://gitlab.com/wgu-gitlab-environment/student-repos/jcayet5/d600-statistical-data-mining/-/tree/master?ref\_type=heads

**Repository branch history screenshot:**

A screenshot of a computer

Description automatically generated

**B1. Propose one research question that is relevant to a real-world organizational situation that will be answered using logistic regression**

The research question that is relevant to the housing dataset and can be answered using logistic regression is: What Factors Determine if a House is Considered Luxury?

**B2. Define one goal of the data analysis**

One goal of the data analysis to identify the variables that significantly influence whether a house is classified as 'luxury'. This goal enables real estate companies to understand what drives luxury status in homes, allowing them to improve their marketing and sales strategies. This specific goal is within the scope of the scenario and data.

**C1. Identify the dependent and all independent variables that are required to answer the research question and justify them**

I have identified the variables that are required to answer the research question. The dependent variable is IsLuxury, and it serves as the target variable indicating whether a house is classified as luxury (1) or not (0). The independent variables that I will use for the logistic regression model are: SquareFootage, NumBathrooms, NumBedrooms, BackyardSpace, CrimeRate, SchoolRating, AgeOfHome, DistanceToCityCenter, EmploymentRate, PropertyTaxRate, RenovationQuality, LocalAmenities, and TransportAccess. For SquareFootage, larger homes with more square footage are typically considered more luxurious. For NumBathrooms, luxury homes typically have multiple bathrooms. For NumBedrooms, a higher number of bedrooms in a property can be associated with luxury. For BackyardSpace, a spacious backyard can contribute to a property's luxury status. For CrimeRate, luxury homes are typically located in areas with low crime rates. For SchoolRating, homes that have high-quality schools nearby can be classified as luxury. For AgeOfHome, newer homes are often classified as luxury, although historic homes can also be valued. For DistanceToCityCenter, luxury homes are often located in prime urban areas. For EmploymentRate, higher employment rates in the area could correspond with more wealthy neighborhoods. For PropertyTaxRate, higher property tax rates are typically associated with wealthy neighborhoods. For RenovationQuality, homes with high-quality renovations can elevate their status to luxury. For LocalAmenities, homes that have access to amenities like parks, shops, and entertainment are likely to be classified as luxury. For TransportAccess, homes that have convenient access to transportation can make them more desirable. I chose these independent variables because they describe property characteristics, neighborhood attributes, and amenities that can influence whether a house is classified as luxury.

**C2. Describe the dependent variable and all independent variables from part C1 using descriptive statistics and include a screenshot for each**

A screenshot of a computer

Description automatically generated

This is the descriptive statistics for IsLuxury. There are 3,528 properties classified as luxury and 3,472 properties classified as non-luxury. IsLuxury has a count value of 7000, mean value of 0.504, minimum value of 0, maximum value of 1, and a range value of 1 (max - min).

A screenshot of a computer

Description automatically generated

This is the descriptive statistics for SquareFootage. SquareFootage has a count value of 7000, mean value of 1048.94, mode value of 550, minimum value of 550, maximum value of 2874.70, and a range value of 2324 (max - min).

A screenshot of a computer

Description automatically generated

This is the descriptive statistics for NumBathrooms. NumBathrooms has a count value of 7000, mean value of 2.13, mode value of 1, minimum value of 1, maximum value of 5.80, and a range value of 4.80 (max - min).

A screenshot of a computer

Description automatically generated

This is the descriptive statistics for NumBedrooms. NumBedrooms has a count value of 7000, mean value of 3, mode value of 3, minimum value of 1, maximum value of 7, and a range value of 6 (max - min)

A screenshot of a computer

Description automatically generated

This is the descriptive statistics for BackyardSpace. BackyardSpace has a count value of 7000, mean value of 511.50, mode values of 300.08, 418.29, and 516.29, minimum value of 0.39, maximum value of 1631.36, and a range value of 1630.97 (max - min).

A screenshot of a computer

Description automatically generated

This is the descriptive statistics for CrimeRate. CrimeRate has a count value of 7000, mean value of 31.22, mode value of 34.01, minimum value of 0.03, maximum value of 99.73, and a range value of 99.70 (max - min).

A screenshot of a computer

Description automatically generated

This is the descriptive statistics for SchoolRating. SchoolRating has a count value of 7000, mean value of 6.94, mode value of 10, minimum value of 0.22, maximum value of 10, and a range value of 9.78 (max - min).

A screenshot of a computer

Description automatically generated

This is the descriptive statistics for AgeOfHome. AgeOfHome has a count value of 7000, mean value of 46.79, mode values of 18.18 and 19.15, minimum value of 0.01, maximum value of 178.68, and a range value of 178.67 (max - min).

A screenshot of a computer

Description automatically generated

This is the descriptive statistics for DistanceToCityCenter. DistanceToCityCenter has a count value of 7000, mean value of 17.47, mode value of 8.29, minimum value of 0, maximum value of 65.20, and a range value of 65.20 (max - min).

A screenshot of a computer

Description automatically generated

This is the descriptive statistics for EmploymentRate. EmploymentRate has a count value of 7000, mean value of 93.71, mode value of 99.9, minimum value of 72.05, maximum value of 99.90, and a range value of 26.95 (max - min).

A screenshot of a computer

Description automatically generated

This is the descriptive statistics for PropertyTaxRate. PropertyTaxRate has a count value of 7000, mean value of 1.50, mode value of 1.43, minimum value of 0.01, maximum value of 3.36, and a range value of 3.35 (max - min).

A screenshot of a computer

Description automatically generated

This is the descriptive statistics for RenovationQuality. RenovationQuality has a count value of 7000, mean value of 5.00, mode value of 10.0, minimum value of 0.01, maximum value of 10.00, and a range value of 9.99 (max - min).

A screenshot of a computer

Description automatically generated

This is the descriptive statistics for LocalAmenities. LocalAmenities has a count value of 7000, mean value of 5.93, mode value of 10.0, minimum value of 0, maximum value of 10, and a range of 10 (max - min).

A screenshot of a computer

Description automatically generated

This is the descriptive statistics for TransportAccess. TransportAccess has a count value of 7000, mean value of 5.98, mode value of 10.0, minimum value of 0.01, maximum value of 10.0, and a range value of 9.99 (max - min).

**C3. Generate univariate and bivariate visualizations of the dependent and independent variables from part C1, including the dependent variable in the bivariate visualizations**

Univariate Visualizations

A screenshot of a graph

Description automatically generated

This countplot visualization shows the distribution of the IsLuxury variable. This indicates that the dataset has an almost equal split between luxury and non-luxury homes. There is just a slightly higher count of luxury homes compared to non-luxury homes. This balanced distribution fits perfectly for logistic regression because there is no significant bias toward one class over another.

A graph with numbers and lines

Description automatically generated with medium confidence

This histogram shows the distribution of the SquareFootage variable in the housing data. The distribution is right-skewed. There is a significant peak at the leftmost bar, indicating that there is a significant number of houses in the data that have a square footage of around 500-600. There are fewer houses with significantly larger square footage.

A graph of a number of bathrooms

Description automatically generated

This histogram shows the distribution of the NumBathrooms variable in the housing data. The distribution is right-skewed. There is a significant peak at the leftmost bar, indicating that there is a significant number of houses in the data that have only 1 bathroom. There are fewer houses with 3 or more bathrooms.

A graph with blue lines and a blue line

Description automatically generated

This histogram shows the distribution of the NumBedrooms variable in the housing data. The distribution is multi-modal because it has multiple peaks. Most houses in the data have 2-4 bedrooms, with 3-bedroom houses being the most common. Houses with 4 or more bedrooms are not very common.

A graph of a number of objects

Description automatically generated with medium confidence

This histogram shows the distribution of the BackyardSpace variable in the housing data. The distribution is slightly right skewed. Most houses in the data have a backyard space of around 400-600, while there are fewer houses with a backyard space above 1000.

A graph of a number of data

Description automatically generated with medium confidence

This histogram shows the distribution of the CrimeRate variable in the housing data. The distribution is slightly right skewed. Most houses in the data are in areas with a crime rate of around 20-50, while there are fewer houses that are located in areas with a crime rate above 60.

A graph with blue lines

Description automatically generated

This histogram shows the distribution of the SchoolRating variable in the housing data. The distribution is left-skewed. There is a significant peak at the rightmost bar, indicating that there is a significant number of houses in the data that have a great school system. There are very few houses that have a poor school system.

A graph of a number of data

Description automatically generated with medium confidence

This histogram shows the distribution of the AgeOfHome variable in the housing data. The distribution is right-skewed. Most houses in the data are relatively new and around the ages 0-50. Houses that are over 80 years old are not very common.

A graph of a line

Description automatically generated with medium confidence

This histogram shows the distribution of the DistanceToCityCenter variable in the housing data. The distribution is right skewed. Most houses in the data are located near the city center within the range of 0-20 miles. Houses that are located farther away from the city center are not common.

A graph with blue and white lines

Description automatically generated

This histogram shows the distribution of the EmploymentRate variable in the housing data. The distribution is left-skewed. Most houses in the data are in areas with high employment rates. Houses that are in areas that have an employment rate below 85 are rare.

A graph of a tax rate

Description automatically generated

This histogram shows the distribution of the PropertyTaxRate variable in the housing data. The distribution is normally distributed. Most houses in the data have property tax rates of around 1-2. Houses with property tax rates below 1 or above 2 are rare.

A graph of a function

Description automatically generated with medium confidence

This histogram shows the distribution of the RenovationQuality variable in the housing data. The distribution is normally distributed. Most houses in the data have a renovation quality of around 3-7. Houses with renovation quality below 3 or above 7 are uncommon.

A graph of a graph with a line

Description automatically generated with medium confidence

This histogram shows the distribution of the LocalAmenities variable in the housing data. There is a significant peak at the rightmost bar, indicating that there is a significant number of houses with local amenities value of 10. Other than that, the distribution looks normally distributed.

A graph of a number of objects

Description automatically generated with medium confidence

This histogram shows the distribution of the TransportAccess variable in the housing data. The histogram looks normally distributed. Most houses in the data have a transport access value of around 4-8. Houses with a transport access value below 4 are not common.

Bivariate Visualizations

A screenshot of a graph

Description automatically generated

This boxplot shows the SquareFootage distribution by luxury classification. Comparing the medians for both categories, luxury properties have a higher median value of SquareFootage than non-luxury properties. This indicates that properties with larger square footage are typically classified as luxury.

A screenshot of a graph

Description automatically generated

This boxplot shows the NumBathrooms distribution by luxury classification. Comparing the medians for both categories, luxury properties have a higher median value of NumBathrooms than non-luxury properties. This indicates that properties with higher number of bathrooms are typically classified as luxury.

A screenshot of a graph

Description automatically generated

This boxplot shows the NumBedrooms distribution by luxury classification. Comparing the distributions for both categories, luxury properties have higher values of NumBedrooms than non-luxury properties. This indicates that properties with higher number of bedrooms are typically classified as luxury.

A screenshot of a computer screen

Description automatically generated

This boxplot shows the BackyardSpace distribution by luxury classification. Comparing the medians for both categories, luxury properties have a slightly higher median value of BackyardSpace than non-luxury properties. This indicates that properties with bigger backyard spaces are typically classified as luxury.

A screenshot of a graph

Description automatically generated

This boxplot shows the CrimeRate distribution by luxury classification. Comparing the medians for both categories, luxury properties have a slightly lower median value of CrimeRate than non-luxury properties. This indicates that properties located in areas with lower crime rates are typically classified as luxury.

A screenshot of a computer screen

Description automatically generated

This boxplot shows the SchoolRating distribution by luxury classification. Comparing the medians for both categories, luxury properties have a higher median value of SchoolRating than non-luxury properties. This indicates that properties located in areas with good school systems are typically classified as luxury.

A screenshot of a graph

Description automatically generated

This boxplot shows the AgeOfHome distribution by luxury classification. Comparing the medians for both categories, luxury properties have a slightly lower median value of AgeOfHome than non-luxury properties. This indicates that properties that were recently built are typically classified as luxury.

A screenshot of a graph

Description automatically generated

This boxplot shows the DistanceToCityCenter distribution by luxury classification. Comparing the medians for both categories, luxury properties have a slightly lower median value of DistanceToCityCenter than non-luxury properties. This indicates that properties located closer to city centers are typically classified as luxury.

A screenshot of a graph

Description automatically generated

This boxplot shows the EmploymentRate distribution by luxury classification. Comparing the medians for both categories, luxury properties have a slightly higher median value of EmploymentRate than non-luxury properties. This indicates that properties located in areas with higher employment rates are typically classified as luxury.

A screenshot of a graph

Description automatically generated

This boxplot shows the PropertyTaxRate distribution by luxury classification. Comparing the medians for both categories, luxury properties have a slightly lower median value of PropertyTaxRate than non-luxury properties. This indicates that properties with lower property tax rates are typically classified as luxury.

A screenshot of a graph

Description automatically generated

This boxplot shows the RenovationQuality distribution by luxury classification. Comparing the medians for both categories, luxury properties have a higher median value of RenovationQuality than non-luxury properties. This indicates that properties with higher renovation quality are typically classified as luxury.

A screenshot of a graph

Description automatically generated

This boxplot shows the LocalAmenities distribution by luxury classification. Comparing the medians for both categories, luxury properties have a higher median value of LocalAmenities than non-luxury properties. This indicates that properties with more amenities nearby are typically classified as luxury.

A screenshot of a computer screen

Description automatically generated

This boxplot shows the TransportAccess distribution by luxury classification. Comparing the medians for both categories, luxury properties have a higher median value of TransportAccess than non-luxury properties. This indicates that properties with more transportation access are typically classified as luxury.

**D1. Split the data into two datasets, with a larger percentage assigned to the training set and a smaller percentage assigned to the test dataset. Provide the files**

A screenshot of a computer program

Description automatically generated

In this code, I stored the target variable IsLuxury into the y variable, and I stored the 13 independent variables into the X variable. I used the train\_test\_split() function from the sklearn library to split the data into training and test sets. 80% of the data goes to training, while 20% of the data goes to test. I have also included the random\_state parameter to ensure reproducibility. The X\_train and y\_train variables are the training data, and X\_test and y\_test variables are the test data. I have concatenated the two training data and stored them into the train\_data variable. I have also concatenated the two test data and stored them into the test\_data variable. Both the variables train\_data and test\_data have been saved as CSV files and exported for submission.

**D2. Perform a logistic regression model using the training set. Optimize the regression model and provide a screenshot of the summary of the model or the key parameters**

Before optimization, check the initial model’s results

A screenshot of a computer

Description automatically generated

We will optimize the logistic regression model using the backward stepwise elimination method. The backward stepwise elimination method iteratively removes non-significant features that have a p-value of greater than 0.05 until only significant features remain. Before we optimize the model, we will look at its summary results to see how the model looks like before the optimization. We're still using the y variable that contains the dependent variable IsLuxury, and the X variable that contains the 13 independent variables. I used the add\_constant() function from the statsmodels library to add a constant to the X variable. Then, I used the train\_test\_split() function to split the data into training and test sets. The training set gets 80% of the data, while the test set gets 20% of the data. After splitting, I used the Logit() function to create and train the logistic regression model on the training data.

Using the summary() function, we can see the logistic regression model's results. The model has a pseudo R-squared value of 0.236, which means the model explains about 23.6% of the variance in the dependent variable. This percentage is very low but not uncommon for logistic regression. For coefficient estimates, the independent variables that have positive coefficient values are SquareFootage, NumBathrooms, NumBedrooms, CrimeRate, RenovationQuality, and LocalAmenities. These variables increase the odds of the house being classified as luxury when they increase. The independent variables that have negative coefficient values are BackyardSpace, SchoolRating, AgeOfHome, DistanceToCityCenter, EmploymentRate, PropertyTaxRate, and Transport Access. These variables decrease the odds of the house being classified as luxury when they increase. As for the p-values, the independent variables that have p-values less than 0.05 are SquareFootage, NumBathrooms, NumBedrooms, PropertyTaxRate, RenovationQuality, and LocalAmenities. These variables are statistically significant predictors of the dependent variable IsLuxury, and they are likely to remain in the model after the model gets optimized. The independent variables with p-values more than 0.05 are BackyardSpace, CrimeRate, SchoolRating, AgeOfHome, DistanceToCityCenter, EmploymentRate, and TransportAccess. These variables do not have a statistically significant relationship with the dependent variable IsLuxury, so they will likely be removed during the optimization process.

Before optimizing the model, its AIC and BIC values have been retrieved. The model has an AIC value of 5958.73, and a BIC value of 6051.56. The AIC measures how well the model fits the data, while penalizing the complexity of the model. The BIC works the same way as AIC but applies a bigger penalty for models with more parameters than AIC does. Since the model has low values of AIC and BIC, it indicates that the model has achieved a balance between fit and simplicity. We will compare these values to the optimized model's AIC and BIC values once the model has been optimized.

**Optimizing the model using the Backward Stepwise Elimination Method**

A screenshot of a computer

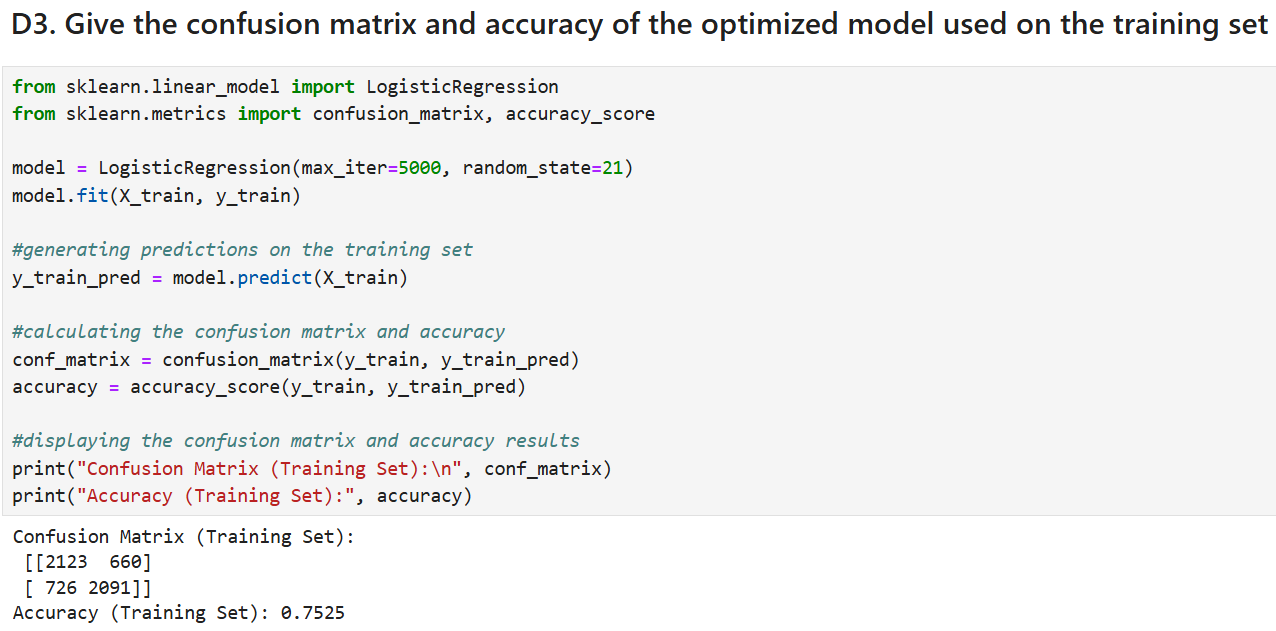
Description automatically generated

In this code, I performed the backward stepwise elimination method to optimize the logistic regression model. The goal of the method is to iteratively remove non-significant variables with p-values greater than 0.05 from the model and retain the statistically significant variables with p-values less than 0.05 in the model. First, I set the value of the variable sig\_level to 0.05, as this variable represents the p-value threshold. Variables that have p-values greater than this threshold will be removed iteratively. Then, I created a while loop that performs the backward stepwise elimination method. Inside the loop, the pvalues() function retrieves the p-values of the variables from the model, then checks to see if the variable with the highest p-value is greater than the 0.05 significance level. If it is, then that variable gets removed from the training and test sets. The model gets refitted with the updated set of variables. The loop continues until all variables have p-values below 0.05. After the backward stepwise elimination loop completes, the final optimized model summary is displayed by using the summary() function.

We'll go over some key metrics from this optimized logistic regression model results. The optimized model has a pseudo R-squared value of 0.235, which means the model explains 23.5% of the variance in the target variable. The optimized model has about the same pseudo R-squared value as the model without the optimization. Regarding coefficient estimates, the independent variables with positive coefficient values are SquareFootage, NumBathrooms, NumBedrooms, RenovationQuality, and LocalAmenities. When these variables increase, they also increase the likelihood of the house being classified as luxury. The only independent variable with a negative coefficient value is PropertyTaxRate, and when this variable increases, it decreases the likelihood of the house being classified as luxury. As for the p-values, the independent variables that were kept in the optimized model are SquareFootage, NumBathrooms, NumBedrooms, PropertyTaxRate, RenovationQuality, and LocalAmenities because these variables have p-values below 0.05 and are statistically significant predictors of the dependent variable IsLuxury. The independent variables that were removed during the backward stepwise elimination process are DistanceToCityCenter, CrimeRate, EmploymentRate, TransportAccess, AgeOfHome, BackyardSpace, and SchoolRating because they have p-values greater than 0.05 and are considered non-significant predictors.

After using the aic() and bic() functions to retrieve the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) values from the optimized model, the AIC and BIC values that we get are 5949.37 and 5995.78 respectively. The AIC measures how well the model fits the data, while penalizing the complexity of the model. Since the optimized model has a low AIC value, it indicates that the model fits the data and manages complexity reasonably well. The BIC is similar to AIC but applies a larger penalty for models with more parameters than AIC does. Since the optimized model has a low BIC value, it indicates that the model fits the data well and is simple. The AIC and BIC values of the optimized model are also slightly lower than the previous model's AIC and BIC values, which means the optimization improved the final model's fit to the data. Overall, the optimized logistic regression model is balanced in terms of fit and complexity, as the AIC and BIC values suggest.

**D3. Give the confusion matrix and accuracy of the optimized model used on the training set**



In this code, I started by using the LogisticRegression() function to initialize a logistic regression model. Inside the function, I set the max\_iter parameter to a value of 5000, which means the model's optimization algorithm will try up to 5000 iterations to find the optimal solution for the model. I also set a random seed for reproducibility. Then, the model is trained using the training data. After training, the optimized model generates predictions on the training set, and the predictions are stored in the variable y\_train\_pred. I used the confusion\_matrix() function to generate a confusion matrix where it provides a breakdown of correct and incorrect predictions by the optimized model. I also used the accuracy\_score() function to display the proportion of correct predictions by the model.

Looking at the optimized model's confusion matrix, the model performed reasonably well in general at correctly classifying both luxury and non-luxury houses in the training set. True negatives have a value of 2123, which means the model correctly classified non-luxury houses as non-luxury. False positives have a value of 660, which means the model incorrectly classified non-luxury houses as luxury. False negatives have a value of 726, which means the model incorrectly classified luxury houses as non-luxury. True positives have a value of 2091, which means the model correctly classified luxury houses as luxury. Despite some misclassifications as seen from both the false positives and false negatives values, the optimized model is fairly effective at accurately identifying both luxury and non-luxury houses in the training data, given the relatively high values of both true positives and true negatives.

The optimized logistic regression model has an accuracy value of 0.75, indicating that the model correctly classifies houses as either luxury or non-luxury 75% of the time. This is a good accuracy score for the optimized model, as it suggests that the model performs reasonably well in distinguishing between luxury and non-luxury houses on the training data.

**D4. Run the prediction on the test dataset using the optimized regression model to evaluate the performance of the model on the rest data based on confusion matric and accuracy. Provide screenshot of the results**

A screenshot of a computer

Description automatically generated

In this code, I used the predict() function for the model to make predictions on the test set. Then, I used the confusion\_matrix() function to create a confusion matrix that displays the counts of true positives, true negatives, false positives, and false negatives from the predictions of the model. I also used the accuracy\_score() function to calculate the proportion of correct predictions by the model.

From the confusion matrix, the value of true negatives is 520, the value of false positives is 169, the value of false negatives is 191, and the value of true positives is 520. The relatively high counts of both true positives and true negatives indicate that the model is reasonably effective at correctly classifying both luxury and non-luxury houses in the test set. However, the model's predictions still have room for improvement given its significant misclassification rates.

The optimized logistic regression model has an accuracy value of 0.74, which means the model accurately identifies houses as either luxury or non-luxury 74% of the time. Since the accuracy value on the test set is lower than the accuracy value on the training set by only 1%, it indicates that the model performs well and generalizes effectively to unseen data. The small difference in accuracy between the training and test sets means that overfitting is not a significant issue. Overall, this is a desirable accuracy value on the test set for the optimized logistic regression model.

**E1. List the packages and libraries chosen for Python and justify them**

The libraries that I have used throughout the logistic regression analysis are Pandas, Matplotlib, Seaborn, Scikit-learn, and Statsmodels. I used the Pandas library primarily to handle the housing dataset that is in CSV format. For example, the function read\_csv() was used to read the housing.csv file and load its data into a Pandas dataframe. The describe() function from Pandas was also very useful, as it provided descriptive statistics like mean, min, and max for each independent variable in the data. I used the Matplotlib library to customize the histograms and boxplots in the univariate and bivariate visualization sections. The functions from Matplotlib allowed me to change the titles, labels, and sizes of the visualizations, which made the visualizations more informative. The Seaborn library is another useful visualization library that is typically used with Matplotlib. The Seaborn library allowed me to generate visually pleasing histograms and boxplots with the help of its histplot() and boxplot() functions. The Scikit-learn is the most important library in the whole analysis. It is a machine learning library that has many tools for developing and examining models. One of the functions from Scikit-learn that I used is the train\_test\_split() function and it was responsible for splitting the housing dataset into training and test sets. Another important function from Scikit-learn is the LogisticRegression() function, and I used it to create and fit a logistic regression model that can accurately classify houses as either luxury or non-luxury. We were also able to evaluate the model's performance and its accuracy on classifying luxury and non-luxury houses by using the functions confusion\_matrix() and accuracy\_score() from Scikit-learn. Lastly, I used the Statsmodels library to set up and run the logistic regression model we created. The function add\_constant() from Statsmodels added an intercept term to the model's predictors, which helped the model fit the housing data more accurately. The function Logit() from Statsmodels ensured that the logistic regression model's output values are between 0 and 1.

**E2. Discuss the method used to optimize the model**

The method I used to optimize the logistic regression model is the backward stepwise elimination method. This method involves iteratively removing non-significant variables with p-values above 0.05 from the model and retaining only the significant variables with p-values below 0.05. By removing the insignificant variables from the model, it enhances the model's performance and simplicity. I created a loop from earlier that performs the backward stepwise elimination process. The process starts by retrieving the p-values of all independent variables in the model, then identifying the variable with the highest p-value. If the variable's p-value is greater than the 0.05 significance level, it suggests that the variable does not significantly contribute to predicting whether a house is classified as luxury, and it is therefore removed from the model. After the variable is removed, the model is re-fitted using the remaining variables. This process repeats until all remaining variables have p-values below 0.05, indicating that they are significant contributors to the classification of a house as luxury. From our final optimized logistic regression model, the independent variables that were removed during the backward stepwise elimination process are DistanceToCityCenter, CrimeRate, EmploymentRate, TransportAccess, AgeOfHome, BackyardSpace, and SchoolRating because they had p-values above 0.05, and were non-significant predictors. The independent variables that remained in the optimized model are SquareFootage, NumBathrooms, NumBedrooms, PropertyTaxRate, RenovationQuality, and LocalAmenities because they had p-values below 0.05, and were statistically significant predictors of the dependent variable IsLuxury.

**E3. Justify the approach discussed in part E2 that was used to optimize the model**

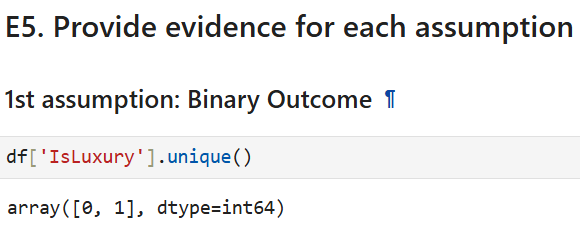
There are a couple of advantages for applying the backward stepwise elimination method to the logistic regression model. One advantage is it simplifies the model by removing irrelevant or non-significant predictors, making the model simpler and easier to understand. Another advantage is it lowers the risk of overfitting since only statistically significant predictors of the dependent variable are retained, improving the model's performance on unseen data. Before optimizing our logistic regression model, the model's AIC and BIC values were 5958.73 and 6051.56 respectively. After the optimization, these values decreased to 5949.37 for AIC and 5995.78 for BIC. A slight decrease in AIC and BIC values indicates that the backward stepwise elimination method that we used to optimize the logistic regression model improved the model's fit to the data and made the model simpler and more interpretable.

**E4. Summarize at least four assumptions of logistic regression**

There are four logistic regression assumptions that must be met to ensure the accuracy and validity of the model. The first assumption is that the dependent variable in the logistic regression model should be binary where it has only two possible outcomes. Our logistic regression model's dependent variable, IsLuxury, has only two unique values [0, 1]. Therefore, the assumption of a binary outcome in our logistic regression has been met. The second assumption is that there should be no multicollinearity in the logistic regression model. Multicollinearity occurs when there are high correlations among the independent variables in the model, leading to unreliable estimates of the regression coefficients. To check for multicollinearity, the Variance Inflation Factor (VIF) will be used to detect the severity of multicollinearity among independent variables. The VIF is "equal to the ratio of the overall model variance to the variance of a model that includes only that single independent variable" (Leung, 2021, par. 31). A VIF value of 1 means there is no correlation with other independent variables, while a VIF value that exceeds 5 or 10 indicates high multicollinearity. After calculating the VIF values for the independent variables in our logistic regression model, we found that all the variables have a VIF value close of 1, indicating that there is no multicollinearity, and the variables are independent of each other. This satisfies the assumption of no multicollinearity in our logistic regression model. The third assumption is that each observation in the dataset should be independent of the others. This means that each observation should be unique and not related to other observations in a way that could influence the outcome. In our case, each row in the housing dataset should represent a unique house and no houses are represented multiple times. I used the duplicated() function from Pandas to check for duplicate rows in the housing dataset and confirmed that no duplicates were found. I also used the Durbin-Watson test to help verify the independence of observations in our logistic regression model. The Durbin-Watson test checks for autocorrelation in the residuals of a regression model. After running the Durbin-Watson test, the value turns out to be 2.02. This value is close to 2, indicating that there is no autocorrelation in the residuals, which satisfies the assumption of independent observations. The duplicated() function and the Durbin-Watson test verified that our logistic regression model satisfies the assumption of independent observations. The fourth assumption is that there should be a large sample size in the logistic regression model. A common guideline is to have at least 10 observations per outcome category for each predictor variable in the model. Since we have 13 predictor variables in our model, we need to have at least 130 observations per category. After using the value\_counts() function to the target variable IsLuxury, we found that there are 3528 luxury and 3472 non-luxury observations, exceeding the requirement of having at least 130 observations per category. Therefore, the large sample size assumption is met.

**E5. Provide evidence that the assumptions were verified by providing screenshots**

1st Assumption



The first assumption is that the target variable should be binary where it has two outcomes. After using the unique() function to our logistic regression's target variable IsLuxury, we found that the variable only has two unique values, 0 and 1. This satisfies the binary outcome assumption.

2nd Assumption

A screenshot of a computer

Description automatically generated

The second assumption is that there should be an absence of multicollinearity in a logistic regression model. Multicollinearity occurs when there are high correlations among the independent variables in the model, which can lead to unstable estimates of the regression coefficients. The Variance Inflation Factor (VIF) will be used to assess the degree of multicollinearity in the model. A VIF value of 1 means there is no multicollinearity, while a VIF value that is greater than 5 or 10 indicates that there is high multicollinearity. In the code, I selected the 13 predictor variables and stored them into variable X, and used the add\_constant() function to add a constant term to X. Then, I created an empty DataFrame to store the names of the predictors and their VIF values. I assigned the predictor names to a column named Feature, and their VIF values to a column named VIF. From the table, all independent variables have a VIF value close to 1, which means there is an absence of multicollinearity, and the variables are independent of each other. Therefore, the assumption of no multicollinearity has been met.

3rd Assumption

A screenshot of a computer program

Description automatically generated

The third assumption is that each observation in the dataset must be independent of the others. There should be no duplicates in the dataset, as it indicates dependencies. In our case, each row in the housing dataset should represent a unique house and no houses are represented multiple times. After using the duplicated() function to check for duplicate rows in the housing dataset, we found that there are no duplicates. I also used the Durbin-Watson test to help verify the independence of observations in our logistic regression model. I started by initializing and fitting a logistic regression model. Then, I used the model to generate prediction probabilities for the positive category in the target variable IsLuxury. After that, I retrieved the residual values after calculating the difference between actual values and predicted probabilities. Then, I ran the Durbin-Watson test on the residuals. The value turns out to be 2.02, which is close to 2 indicating that there is no autocorrelation in the residuals, which satisfies the assumption of independent observations. The duplicated() function and the Durbin-Watson test verified that our logistic regression model satisfies the assumption of independent observations.

4th Assumption

A screenshot of a computer

Description automatically generated

The fourth assumption is there should be a large sample size in a logistic regression model. A common guideline is to have at least 10 observations per outcome category for each predictor variable in the model. Since we have 13 predictor variables in our model, we must have a minimum of 130 observations per category. After using the value\_counts() function to the dependent variable IsLuxury, we found that there are 3528 luxury and 3472 non-luxury observations, which are far greater than the required minimum of 130 observations per category. Therefore, the large sample size assumption is met.

**E6. Provide the regression equation and discuss the coefficient estimates**

Logistic Regression Equation: *log(P(Luxury)/1 - P(Luxury)) = Intercept + (Coefficient of X1) \* X1 + (Coefficient of X2) \* X2 + ... + (Coefficient of Xn) \* Xn*

Coefficients from our regression output: *log(P(Luxury)/1 - P(Luxury)) = -5.7623 + 0.0018 \* SquareFootage + 0.6153 \* NumBathrooms + 0.6670 \* NumBedrooms - 0.1291 \* PropertyTaxRate + 0.0945 \* RenovationQuality + 0.0460 \* LocalAmenities*

The value -5.7623 is the intercept and it represents the log-odds of a house being classified as luxury when all predictor variables are zero. The intercept serves as the baseline for the logistic regression model. Regarding the coefficient estimates, we will discuss each predictor and its corresponding coefficient estimate. The SquareFootage predictor has a coefficient value of 0.0018, which indicates that for each additional square foot, the log-odds of a house being classified as luxury increases by 0.0018. The NumBathrooms predictor has a coefficient value of 0.6153, which indicates that for each additional bathroom, the log-odds of a house being classified as luxury increases by 0.6153. The NumBedrooms predictor has a coefficient value of 0.6670, which indicates that for each additional bedroom, the log-odds of a house being classified as luxury increases by 0.6670. The PropertyTaxRate predictor has a coefficient value of -0.1291, which indicates that for each one-unit increase in the property tax rate, the log-odds of a house being classified as luxury decreases by 0.1291. The RenovationQuality predictor has a coefficient value of 0.0945, which indicates that for each one-unit increase in renovation quality, the log-odds of a house being classified as luxury increases by 0.0945. The LocalAmenities predictor has a coefficient value of 0.0460, which indicates that for each one-unit increase in local amenities, the log-odds of a house being classified as luxury increases by 0.0460.

To summarize, the predictors with positive coefficient values are SquareFootage, NumBathrooms, NumBedrooms, RenovationQuality, and Local Amenities. These predictors increase the log-odds of a house being classified as luxury when they increase. The only predictor with a negative coefficient value is PropertyTaxRate, and this predictor decreases the log-odds of a house being classified as luxury when it increases.

**E7. Discuss the model metrics (This part has been revised)**

The optimized logistic regression model has an accuracy value of 0.74 on the test set, meaning it correctly classifies approximately 74% of the houses as either luxury or non-luxury. This accuracy suggests that the model performs moderately well in distinguishing between luxury and non-luxury houses, though there is still room for improvement.

The optimized logistic regression model has an accuracy value of 0.75 on the training set, and an accuracy value of 0.74 on the test set. The accuracy value on the training set is only 1% higher than the accuracy value on the test set. This small difference suggests that the model generalizes well to unseen data and is not overfitting. Overfitting typically occurs when the accuracy value on the training set is significantly higher than on the test set, but in this case, the accuracies are almost similar to each other, indicating a well-balanced model. Overall, the model performs reasonably well in predicting whether a house is luxury or non-luxury based in its accuracy values on both training and test sets.

The confusion matrix for the training set has true negatives value of 2123, false positives value of 660, false negatives value of 726, and true positives of 2091. The confusion matrix for the test set has true negatives value of 520, false positives value of 169, false negative value of 191, and true positives value of 520. In the training set, the model accurately predicted 4214 out of 5600 observations. In the test set, the model accurately predicted 1040 out of 1400 observations. The model appears to perform consistently across both datasets because the ratio of true positives and true negatives is similar on both sets. The model's error rate is also consistent across both datasets because the ratio of false positives and false negatives is similar between the two sets. Similar rates of accurate and inaccurate classifications in both sets indicate that the model generalizes well to unseen data without overfitting, which is ideal for a model that predicts whether a house is luxury or non-luxury.

The pseudo R-squared value from the optimized model is 0.2354. This value is lower than the initial model's R-squared value which is 0.2360. This indicates that the backward stepwise elimination process effectively simplified the logistic regression model without sacrificing much of its predictive power. This trade-off is reasonable, especially if the goal is to achieve a simpler and more practical model.

**E8. Discuss the results and implications of your prediction analysis**

The final optimized model's similar accuracy on both training and test sets suggests that the model generalizes well to new data and shows no signs of overfitting. Additionally, the balanced distribution of true positives, true negatives, false positives, and false negatives across both sets indicates that the model is dependable and effective at accurately classifying luxury and non-luxury houses. This strong performance was achieved by retaining only the statistically significant variables with p-values below 0.05 through backward stepwise elimination. The significant variables were SquareFootage, NumBathrooms, NumBedrooms, PropertyTaxRate, RenovationQuality, and LocalAmenities, and these variables play a crucial role in the model's ability to predict whether a house is luxury or non-luxury. The variables with positive coefficients are SquareFootage, NumBathrooms, NumBedrooms, RenovationQuality, and LocalAmenities, and they increase the likelihood of a house being classified as luxury. The only variable with a negative coefficient is PropertyTaxRate, and it decreases the likelihood of a house being classified as luxury. The optimization process clearly improved the model's fit while making it simpler and more interpretable, as indicated by the reduced AIC and BIC values. Regarding the model's assumptions, it was confirmed that the assumptions such as binary outcome, absence of multicollinearity, independence of observations, and large sample size have been met by our optimized logistic regression model.

**E9. Recommend a course of action for the real-world organizational situation based on your results and implications**

Based on the results and implications of our logistic regression model, a course of action that organizations like real estate companies or developers can perform is to focus on key predictors that drive luxury property classification when making marketing strategies. Organizations should emphasize key features like SquareFootage, NumBathrooms, NumBedrooms, RenovationQuality, and LocalAmenities when marketing properties to buyers that are interested in luxury homes, as luxury homebuyers tend to prioritize such features. Organizations should also invest in properties with ample square footage, multiple bathrooms/bedrooms, high-quality renovations, good local amenities, and lower property tax rates because these properties can be valued higher and sold, leading to greater profitability.

**F. Panopto Video**

Here’s the link to my Panopto Presentation:

<https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=d4a4666f-9b26-4645-84b7-b2290161f65c>

**References**

Leung, K. (2021, October 4). *Assumptions of Logistic Regression, Clearly Explained*. Medium. https://towardsdatascience.com/assumptions-of-logistic-regression-clearly-explained-44d85a22b290