D600 – Statistical Data Mining

Performance Assessment #2 – Logistic Regression Analysis

Bernard Connelly

Master of Science, Data Analytics, Western Governors University

Dr. Keiona Middleton

January 27, 2025

D600 – Statistical Data Mining: Logistic Regression Analysis

Purpose of Analysis

B1. Research Question

Using the provided dataset, the metrics behind determining if a home is classified as a "luxury" home would be helpful to an organization in maximizing the price and marketability of a house. To that end, the question "Do the variables crime_rate, school_rating, backyard_space, local_amenities, fireplace, and garage contribute to a house being classified as a luxury?" is the working research question for this assignment.

B2. Goal of Analysis

By running a logistic regression model on the characteristics identified in the research question, an organization can narrow down which variables are more likely predictors of whether a house is identified as "luxury." In doing so, they can ascertain the incremental values to increase each to transition the house into a "luxury" status. Under the assumption that "luxury" homes are worth more than their alternatives, a company can identify which characteristics to target in their purchases or which characteristics to include in new builds to increase profit and returns on their investments.

Summarization of Data Preparation

C1. Identification of Variables

The independent variables selected for this analysis are crime_rate, school_rating, backyard_space, local_amenities, fireplace, and garage. These will be used to determine their effect on the dependent variable, luxury, via a logistic regression model. Logically, certain variables about the house size, such as the number of rooms or square footage, will directly

correlate with whether or not a house is considered luxury. Many of the selected variables are outside the typical standards for a luxury house, so researching them more directly can assist the organization in determining which characteristics will be most valuable in addition to the home's overall size.

C2. Statistical Description of Variables

Below is a breakdown of the relevant descriptive statistics for the variables in this analysis:

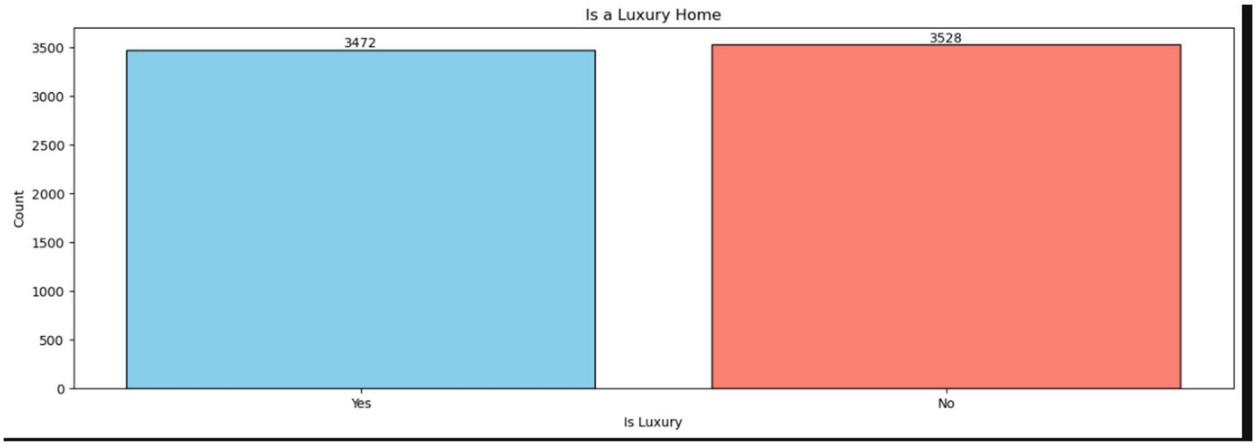
```python
#Variable Descriptions
## C2 - Describe the dependent variable and all independent
print('Descriptive Statistics\n')
print('\nQuantitative Variables\n')#Showing Mean, Standard D
print('\nCrime Rate')
print(df['crime_rate'].describe())
print('\nSchool Rating')
print(df['school_rating'].describe())
print('\nBackyard Space')
print(df['backyard_space'].describe())
print('\nLocal Amenities')
print(df['local_amenities'].describe())
print('\n\nQualitative Variables\n')#Showing frequencies in
print('Luxury') #Dependent Variable
print(df['is_luxury'].value_counts())
print('\nHas a Fireplace')
print(df['fireplace'].value_counts())
print('\nHas a Garage')
print(df['garage'].value_counts())
```

```
Descriptive Statistics


Quantitative Variables


Crime Rate
count    7000.000000
mean       31.226194
std        18.025327
min         0.030000
25%        17.390000
50%        30.385000
75%        43.670000
max        99.730000
Name: crime_rate, dtype: float64

School Rating
count    7000.000000
mean        6.942923
std         1.888148
min         0.220000
25%         5.650000
50%         7.010000
75%         8.360000
max        10.000000
Name: school_rating, dtype: float64
```

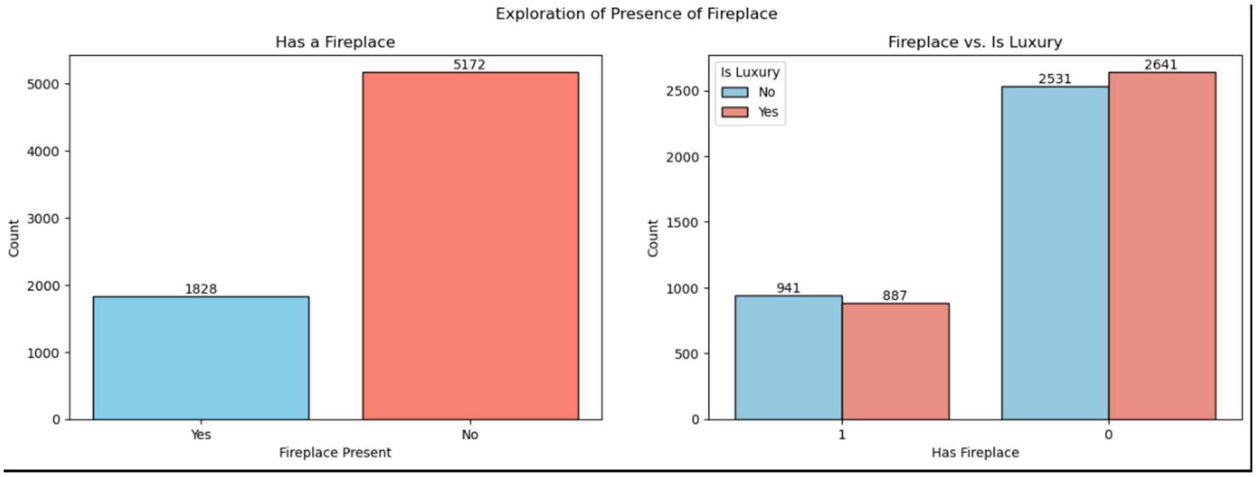
```
Backyard Space
count    7000.000000
mean      511.507029
std       279.926549
min         0.390000
25%       300.995000
50%       495.965000
75%       704.012500
max      1631.360000
Name: backyard_space, dtype: float64

Local Amenities
count    7000.000000
mean        5.934579
std         2.657930
min         0.000000
25%         4.000000
50%         6.040000
75%         8.050000
max        10.000000
Name: local_amenities, dtype: float64


Qualitative Variables

Luxury
is_luxury
1    3528
0    3472
Name: count, dtype: int64

Has a Fireplace
fireplace
0    5172
1    1828
Name: count, dtype: int64

Has a Garage
garage
0    4488
1    2512
Name: count, dtype: int64
```

C3. Statistical Visualizations

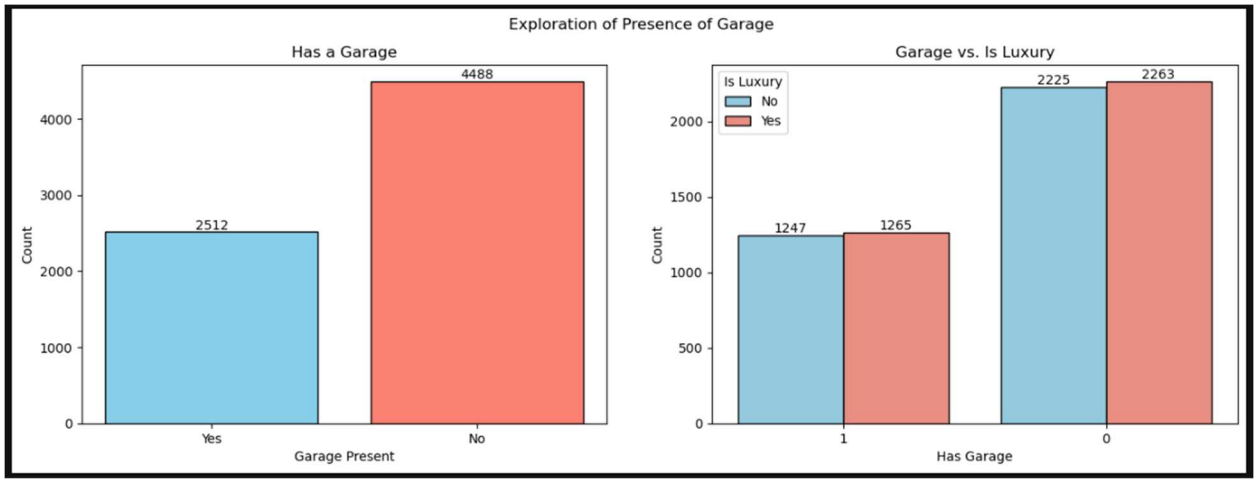
Below are visualizations of the variables selected for this analysis. Included are both univariate representations and bivariate representations that compare the independent variables alongside the dependent variable. The independent variables, alongside other qualitative variables, are shown first, and the quantitative variables are shown last.



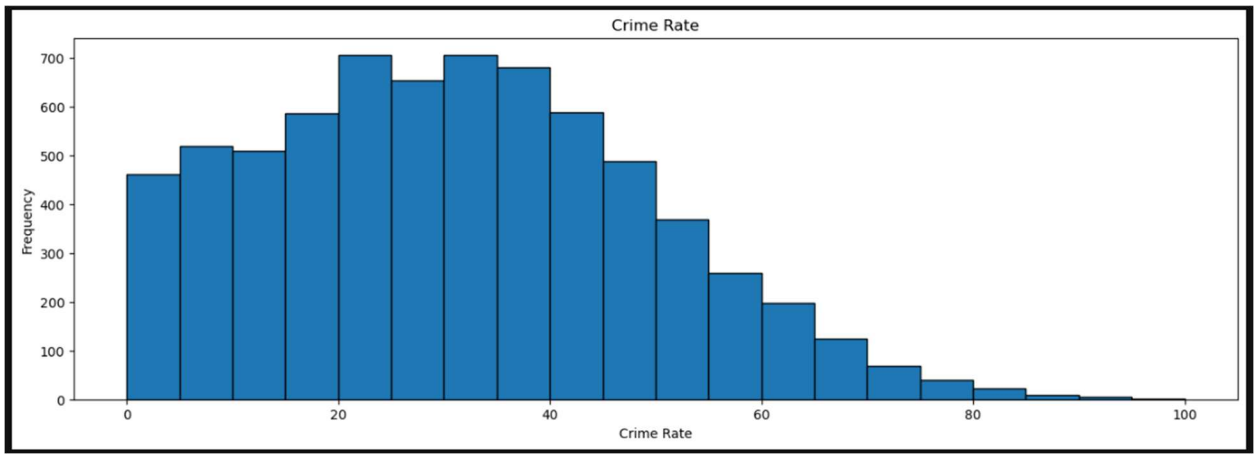
The dependent variable, luxury, is close to an even split in the data.

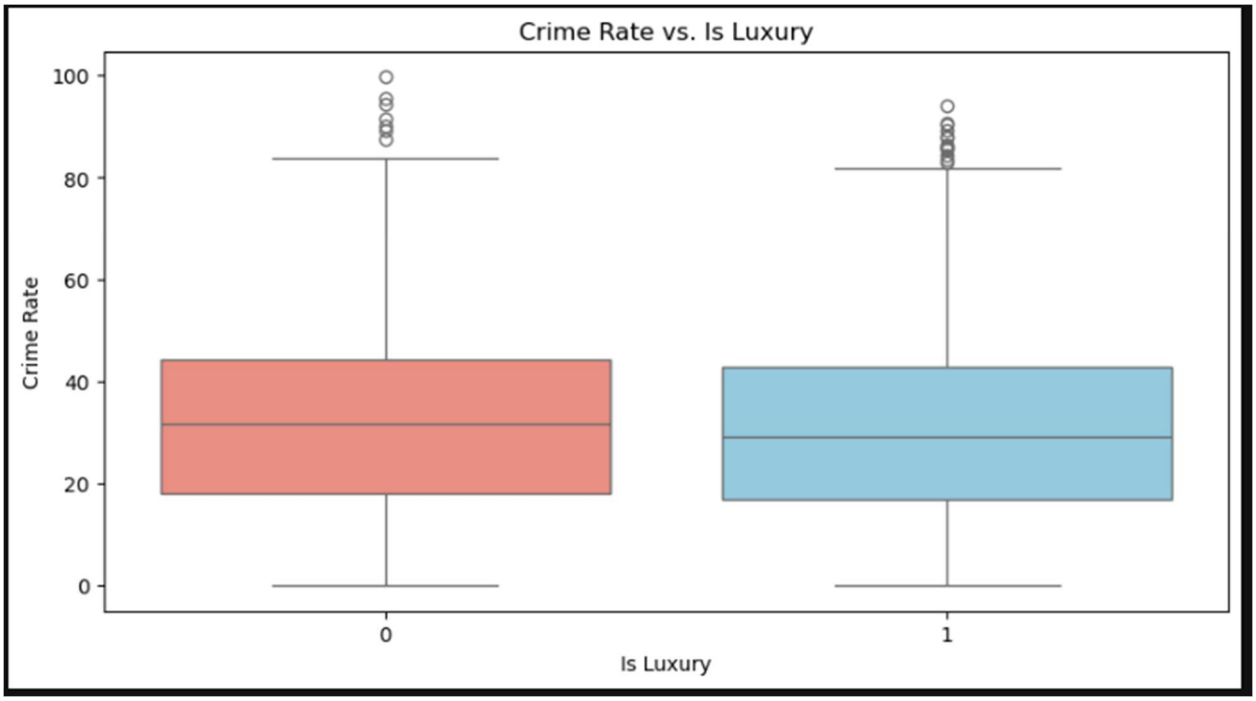


The presence of a fireplace is significantly lower in most houses for the dataset overall, and when compared alongside whether or not a home is a luxury model, there is not a significant skew in one direction or the other.

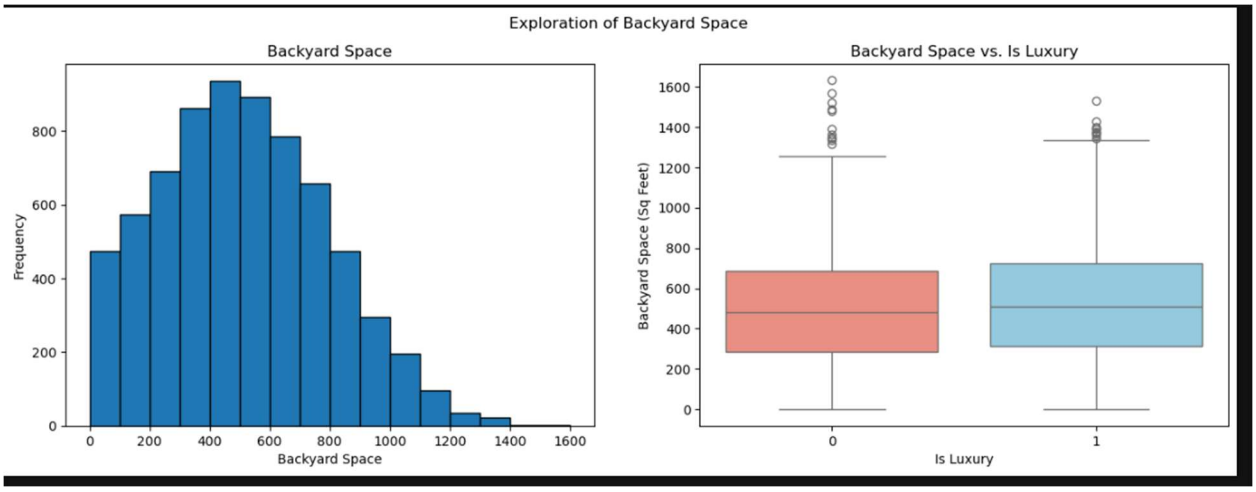


Similar to a fireplace, fewer houses have a garage, but there appears to be little difference between a house having a garage and not having a garage when it is classified as luxury.



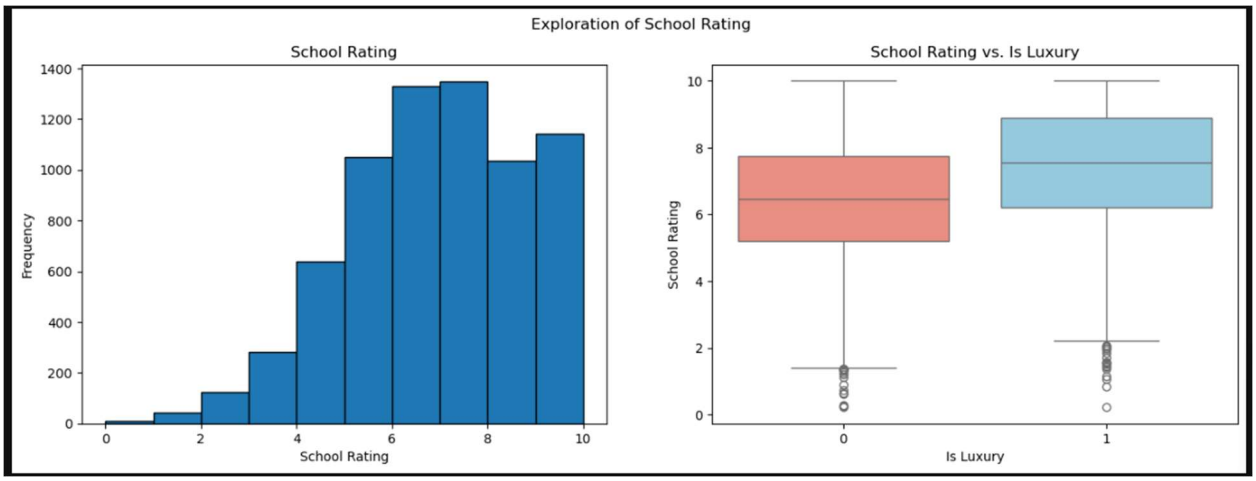


The crime rate is right skewed, as the dataset overall has a higher frequency of houses with lower crime rates than high ones. Compared to luxury homes in the boxplot, there is a slight skew toward lower crime rate areas correlating with luxury homes. Still, the difference is insignificant, and the boxplots are very similar.

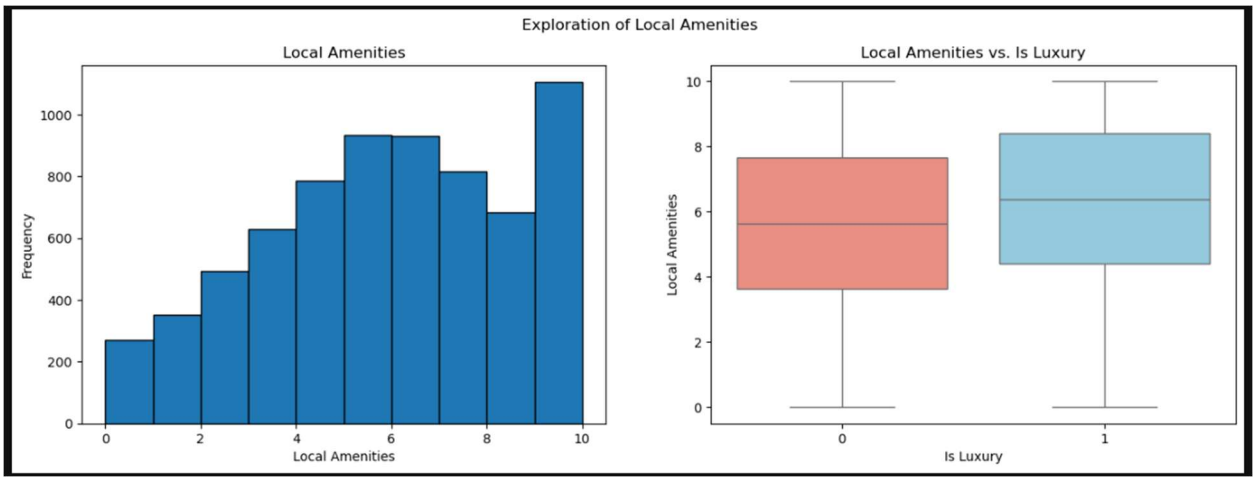


Backyard space demonstrates a reasonably normal distribution, with a significant clustering of houses that have between 0 and 100 feet square feet. Luxury homes favor having

larger backyards on average when looking at the boxplot. However, there are outliers in the non-luxury dwellings with the largest backyards in the dataset.



The dataset favors residences that are located in higher school-rating neighborhoods. Additionally, this variable has a much more apparent effect on whether a house can be considered a luxury, as the data favors those with school ratings.



The data shows a relatively normal distribution of local amenities, with a cluster of values visible at the high end near the 10 value. Similar to school ratings, there is a preference

for higher availability of amenities when considering if a house is regarded as a luxury home.

Logistic Regression Modelling

D1. Creation of Training & Test Data Sets

The data was split with an 80/20 ratio for this analysis, and each dataset was exported correctly to a file. The code for completing this is as follows:

```
#Splitting the Dataset into Test and Training
## D1 - Split the data into two datasets, with a larger percentage assigned to the training dataset and a smaller percentage assigned to the test data set. Provide the files.
y = df.is_luxury
X = df[['crime_rate', 'school_rating', 'backyard_space', 'local_amenities', 'fireplace', 'garage']].assign(const=1)

#Splitting the Dataset into a Test and Training dataset with an 80/20 split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
y_train = y_train.to_frame() #Converting to dataframe for ease in exporting
y_test = y_test.to_frame()
print(f"Training data: {X_train.shape}, Testing data: {X_test.shape}")
print(f"Training labels: {y_train.shape}, Testing labels: {y_test.shape}")
```

```
Training data: (5600, 7), Testing data: (1400, 7)
Training labels: (5600, 1), Testing labels: (1400, 1)
```

```
#Combining the datasets for exporting
train_data = pd.concat([X_train, y_train], axis=1)
test_data = pd.concat([X_test, y_test], axis=1)
```

```
#Exporting the Test & Train Datasets to share
train_data.to_csv("training_data", index=False)
test_data.to_csv("test_data", index=False)
```

D2. Creation & Optimization of the Model

The initial Logistic Regression model was created with no variables removed or optimized, and the output is as follows:

```
logit_model = sm.Logit(y_train, X_train).fit()
print(logit_model.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.653471
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:              is_luxury   No. Observations:                 5600
Model:                          Logit   Df Residuals:                     5593
Method:                           MLE   Df Model:                            6
Date:                Mon, 27 Jan 2025   Pseudo R-squ.:                 0.05713
Time:                        22:29:59   Log-Likelihood:                -3659.4
converged:                       True   LL-Null:                       -3881.2
Covariance Type:            nonrobust   LLR p-value:                1.231e-92
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
crime_rate     0.0020      0.002      1.273      0.203      -0.001       0.005
school_rating  0.2824      0.016     17.767      0.000       0.251       0.314
backyard_space 0.0002   9.97e-05      2.479      0.013    5.18e-05       0.000
local_amenities 0.0707     0.011      6.649      0.000       0.050       0.092
fireplace     -0.1007      0.064     -1.579      0.114      -0.226       0.024
garage         0.0008      0.058      0.014      0.989      -0.113       0.114
const         -2.5141      0.154    -16.371      0.000      -2.815      -2.213
==============================================================================
```

To optimize the model, backward stepwise elimination was initiated using an if/else loop to remove the variables whose p-values were under the threshold of 0.05. Ultimately, three variables were identified as not statistically significant and were removed from the model – garage, fireplace, and crime rate. Below is the code and output of the optimization of the steps and the final model with relevant coefficients. As the statsmodels.api .The Logit class does not automatically output to AIC and BIC values; these were added manually with a separate code block.

```python
#Backwards Stepwise Elimination
logit_model = sm.Logit(y_train, X_train).fit()
print(logit_model.summary())

def backward_elimination(X, y, threshold=0.05): ##Initiating Loop to remove variables based upon p-values
    while True:
        model = sm.Logit(y, X).fit(disp=0)
        p_values = model.pvalues
        max_p = p_values.max()
        if max_p > threshold:
            feature_to_remove = p_values.idxmax()
            print(f"Removing '{feature_to_remove}' with p-value {max_p:.4f}")
            X = X.drop(columns=[feature_to_remove])
        else:
            break

    return X, model

X_train_reduced, final_model = backward_elimination(X_train, y_train)
print(final_model.summary())
#Printing the AIC and BIC Manually
print(f"\nAIC: {final_model.aic:.4f}")
print(f"BIC: {final_model.bic:.4f}")
```

```
Optimization terminated successfully.
         Current function value: 0.653471
         Iterations 5
                           Logit Regression Results
==============================================================================
Dep. Variable:              is_luxury   No. Observations:                 5600
Model:                          Logit   Df Residuals:                     5593
Method:                           MLE   Df Model:                            6
Date:                Mon, 27 Jan 2025   Pseudo R-squ.:                 0.05713
Time:                        22:29:59   Log-Likelihood:                -3659.4
converged:                       True   LL-Null:                       -3881.2
Covariance Type:            nonrobust   LLR p-value:                 1.231e-92
==============================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
crime_rate         0.0020      0.002      1.273      0.203      -0.001       0.005
school_rating      0.2824      0.016     17.767      0.000       0.251       0.314
backyard_space     0.0002   9.97e-05      2.479      0.013    5.18e-05       0.000
local_amenities    0.0707      0.011      6.649      0.000       0.050       0.092
fireplace         -0.1007      0.064     -1.579      0.114      -0.226       0.024
garage             0.0008      0.058      0.014      0.989      -0.113       0.114
const             -2.5141      0.154    -16.371      0.000      -2.815      -2.213
==============================================================================
Removing 'garage' with p-value 0.9891
Removing 'crime_rate' with p-value 0.2030
Removing 'fireplace' with p-value 0.1177
```

The final model and output are:

```
                           Logit Regression Results
==============================================================================
Dep. Variable:              is_luxury   No. Observations:                 5600
Model:                          Logit   Df Residuals:                     5596
Method:                           MLE   Df Model:                            3
Date:                Mon, 27 Jan 2025   Pseudo R-squ.:                0.05661
Time:                        22:29:59   Log-Likelihood:                -3661.5
converged:                       True   LL-Null:                       -3881.2
Covariance Type:            nonrobust   LLR p-value:                6.367e-95
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
school_rating   0.2786      0.016     17.834      0.000       0.248       0.309
backyard_space  0.0002   9.96e-05      2.458      0.014    4.96e-05       0.000
local_amenities 0.0709      0.011      6.673      0.000       0.050       0.092
const          -2.4504      0.132    -18.584      0.000      -2.709      -2.192
==============================================================================

AIC: 7330.9461
BIC: 7357.4682
```

D3. Confusion Matrix & Accuracy (Training Dataset)

Below are the results of the confusion matrix and the accuracy of the optimized model

```
#Confusion Matrix - Training Dataset
## D3 - Give the confusion matrix and accuracy of the optimized model used on the training set.

y_train_pred = (logit_model.predict(X_train) >= 0.5).astype(int)

##Putting into a Dataframe for cleaner visualization
cm = confusion_matrix(y_train, y_train_pred)
cm_training_df = pd.DataFrame(cm,
                    index=["Actual: No (0)", "Actual: Yes (1)"],
                    columns=["Predicted: No (0)", "Predicted: Yes (1)"])

print(cm_training_df)

#Computing Accuracy
accuracy = accuracy_score(y_train, y_train_pred)
print(f"\nAccuracy: {accuracy:.4f}")
```

```
                Predicted: No (0)  Predicted: Yes (1)
Actual: No (0)               1648               1117
Actual: Yes (1)              1024               1811

Accuracy: 0.6177
```

An accuracy of 61.7% demonstrates that the model is trending in the correct direction to predict, and the variables contained within are essential to whether or not a house can be classified as a luxury home. However, there are additional factors that need to be taken into consideration.

D4. Test Dataset Logistic Model, Confusion Matrix, & Accuracy

The model was used on the test dataset to confirm that it generalizes well with another dataset. The results of the analysis are as follows:

```
#Running Optimized Model on Test dataset
## D4 - Run the prediction on the test dataset using the optimized regression model

X_test_trimmed = X_test[["school_rating", "backyard_space", "local_amenities"]] #Select only the relevant variables found from training
logit_model = sm.Logit(y_test, X_test_trimmed).fit()
print(logit_model.summary())

# Print AIC and BIC manually
print(f"\nAIC: {logit_model.aic:.4f}")
print(f"BIC: {logit_model.bic:.4f}")
```

```
Optimization terminated successfully.
         Current function value: 0.689348
         Iterations 4
                           Logit Regression Results
==============================================================================
Dep. Variable:             is_luxury   No. Observations:                 1400
Model:                         Logit   Df Residuals:                     1397
Method:                          MLE   Df Model:                            2
Date:                Mon, 27 Jan 2025   Pseudo R-squ.:                0.005410
Time:                        23:24:41   Log-Likelihood:                -965.09
converged:                       True   LL-Null:                       -970.34
Covariance Type:           nonrobust   LLR p-value:                  0.005250
==============================================================================
                   coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
school_rating      0.0546      0.019      2.886      0.004       0.018       0.092
backyard_space    -0.0003      0.000     -1.757      0.079      -0.001    3.63e-05
local_amenities   -0.0204      0.019     -1.078      0.281      -0.057       0.017
==============================================================================

AIC: 1936.1732
BIC: 1951.9059
```

The dataset only used the trimmed-down variables of school_rating, backyard_space, and local_amenities as a part of the analysis.

```python
#Confusion Matrix for Test Dataset
y_test_pred = (logit_model.predict(X_test_trimmed) >= 0.5).astype(int)

##Putting into a Dataframe for cleaner visualization
cm = confusion_matrix(y_test, y_test_pred)
cm_test_df = pd.DataFrame(cm,
                          index=["Actual: No (0)", "Actual: Yes (1)"],
                          columns=["Predicted: No (0)", "Predicted: Yes (1)"])

print(cm_test_df)

#Computing Accuracy
accuracy = accuracy_score(y_test, y_test_pred)
print(f"\nAccuracy: {accuracy:.4f}")
```

```
                Predicted: No (0)  Predicted: Yes (1)
Actual: No (0)                194                 513
Actual: Yes (1)               130                 563

Accuracy: 0.5407
```

Summary and Results

E1. Libraries Used

The following libraries were used, and their justifications are annotated afterward for reference:

```python
#Importing Relevant Packages
##E1 - list the packages or libraries you have chosen for Python or R and justify how each item on the list supports the analysis.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt #Used for Data Visualizations
import matplotlib.ticker as mticker #Used to properly scale the axes in data plots
import seaborn as sns #Used for Data Visualizations
import statsmodels.api as sm #Used to create the Logistic Regression Model
from statsmodels.stats.outliers_influence import variance_inflation_factor #Used to check Variance Inflation Factor (VIF)
from sklearn.metrics import confusion_matrix #Used to create confusion Matrix
from sklearn.metrics import accuracy_score #Used to calculate the Accuracy attribute
from sklearn.model_selection import train_test_split #Used to split the datasets
from sklearn.model_selection import KFold #Used to cross-validate with Kfold
```

E2 + E3 Method of Optimization & Justification of Approach

Backward Stepwise Elimination was utilized to optimize the model. This approach was most relevant to the project as the recommendation was to use six independent variables, and

additional values were not required. Backward stepwise elimination allowed me to assess the individual variables on their statistical significance based on their p-values and remove those that did not have a relevant effect on the model. Additionally, by limiting the model to 6 initial variables and trimming it down further based on statistical significance, I ensured it did not become too complex or bloated, resulting in poor generalization to new data sets.

Backward Stepwise Elimination is also preferable over forward selection when there are not too many predictors in the dataset. This was the case with the housing dataset presented for this project. Additionally, by reducing the variables in the data model, the overall interpretability of the model is increased, as it is much easier to interpret a model with only a handful of variables as opposed to a larger model with upwards of 5 or more.

E4 + E5. Assumptions of Logistic Regression and Verification of Assumptions

Amongst the numerous assumptions of logistic regression, the following are some examples I tested for and verified as a part of the model.

- The dependent variable must be categorical in nature. In this scenario, is_luxury is a categorical variable with responses listed as 1/0.

```
#Demonstrating the Dependent Variable is Categorical
df['is_luxury'].value_counts()
## All values are 1 or 0

is_luxury
1    3528
0    3472
Name: count, dtype: int64
```

- The dataset must have sufficient records for the analysis to be valid. Generally speaking, this would be less than 500 records. As per the below, the dataset has 7,000 observations, well outside the threshold where this would be an issue.

```
#Demonstrating the Dataset has a sufficient sample size
df.shape
## 7000 Records is well above the lower threshold for this to be an issue
```

```
(7000, 22)
```

- Multicollinearity cannot exist amongst the independent variables in the dataset. Otherwise, it becomes increasingly difficult to determine the individual effects of the variables. This would also have a significant impact on the coefficients. Per the screenshot below, the correlation matrix has no values approaching or near one, and all features listed in the VIF were all at or near 1, well within an acceptable range.

```
#Checking for Multicolinnearity with a Correlation Matrix
independent_variables = df[["school_rating", "backyard_space", "local_amenities"]]
corr_matrix = independent_variables.corr()

print(corr_matrix)
## No variables are at or approaching 1
```

```
                school_rating  backyard_space  local_amenities
school_rating        1.000000        0.047320         0.138622
backyard_space       0.047320        1.000000         0.046857
local_amenities      0.138622        0.046857         1.000000
```

```
#Confirming lack of Multicollinearity with Variance Inflation Factor
vif_data = pd.DataFrame()
vif_data["Feature"] = X_train.columns
vif_data["VIF"] = [variance_inflation_factor(X_train.values, i) for i in range(X_train.shape[1])]
print(vif_data)
## No values greater than 1 indicate low levels of multicollinearity
```

```
           Feature        VIF
0       crime_rate   1.038513
1    school_rating   1.057134
2   backyard_space   1.006620
3  local_amenities   1.022715
4        fireplace   1.001304
5           garage   1.001031
6            const  27.665474
```

- Each row must be entirely independent of the other rows, and no duplicate rows can exist within the dataset. Any instances of duplicate values risk affecting the integrity of the model.

```
# Checking for duplicate rows in the dataset
duplicated_values = df.duplicated().sum()
print(duplicated_values)
## No duplicated values indicate all rows are independent of one another
```

```
0
```

E6. Logistic Regression Equation & Coefficient Estimates

The logistic regression equation for this project is as follows: log(P(is_luxury = 1) / (1 - P(is_luxury = 1))) = 0.0546 * school_rating - 0.0003 * backyard_space - 0.0204 * local_amenities. This was derived using the results of the logistic regression model on the test set.

```
Optimization terminated successfully.
         Current function value: 0.689348
         Iterations 4
                           Logit Regression Results
==============================================================================
Dep. Variable:              is_luxury   No. Observations:                 1400
Model:                          Logit   Df Residuals:                     1397
Method:                           MLE   Df Model:                            2
Date:                Tue, 28 Jan 2025   Pseudo R-squ.:                0.005410
Time:                        22:05:14   Log-Likelihood:                -965.09
converged:                       True   LL-Null:                       -970.34
Covariance Type:            nonrobust   LLR p-value:                  0.005250
==============================================================================
                     coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
school_rating      0.0546      0.019      2.886      0.004       0.018       0.092
backyard_space    -0.0003      0.000     -1.757      0.079      -0.001    3.63e-05
local_amenities   -0.0204      0.019     -1.078      0.281      -0.057       0.017
==============================================================================

AIC: 1936.1732
BIC: 1951.9059
```

The coefficients in the final model generally identified that our model has a weak fit and do not strongly explain why a house would be classified as luxury. The variable school_rating had a coefficient of .0546 and a p-value of .004, indicating a statistically significant positive relationship. The variable backyard_space had a coefficient of -.0003 and a p-value of .079,

which means a negative impact but is generally a weak coefficient. It is also not strongly significant as a predictor, as the p-value was greater than the standard .05 threshold. Finally, the variable local_amenities had a coefficient of -.0204 and a p-value of .281. This indicates a statistically insignificant value and would decrease the probability of a home being classified as luxury.

E7. Model Metrics

The accuracy and confusion matrices of the training and test sets were calculated together in one block of code for each. The resultant responses were as follows:

Training Set:

```
                 Predicted: No (0)  Predicted: Yes (1)
Actual: No (0)                1648                1117
Actual: Yes (1)               1024                1811

Accuracy: 0.6177
```

Test Set:

```
                 Predicted: No (0)  Predicted: Yes (1)
Actual: No (0)                 194                 513
Actual: Yes (1)                130                 563

Accuracy: 0.5407
```

The accuracy of the test set identified the model as 54% accurate. The training set, however, identified the model as 61.7% accurate. The difference between these values suggests the model is overfitting or not generalizing well to new data. Additionally, when inspecting the differences between the confusion matrices, there are significant discrepancies between the training and test datasets. The test matrix needs to be scaled up to the training set to accurately

compare the confusion matrices. This can be completed by multiplying the test set by 4, which results in the following confusion matrix:

[776, 2052

520, 2252]

Comparing the training and scaled test confusion matrices resulted in the following outcomes:

- The training set had 872 more true negative values
- The training set had 935 less false positive values
- The training set had 504 more false negative values
- The training set had 441 less true positive values

The above results indicate that the model was more conservative in predicting luxury homes and less aggressive in detecting them. Additionally, the test set had more false positives, further indicating poor generalization. The model itself is too conservative.

E8. Results and Implications

The results and implications of the analysis indicate that the variables selected to identify luxury houses were not effective predictors. The pseudo-R^2 value of .005 indicates less than 1% explanation of variance in the model. Overall, school_rating was the only independent variable that had a net positive correlation with the dependent variable. The coefficient value of .05 indicates a small effect at best. Both backyard space and local_amenities had slightly negative impacts, if any impact at all due to a low p-value for both. The model does not effectively predict what houses could be classified as luxury and does not provide a high degree of predictive power for which characteristics would be best to focus on to achieve that outcome.

E9. Recommended Course of Action

Since the model was not effective in producing predictive power for the is_luxury variable, I recommend the organization forego utilizing the model to provide any predictions. The variable school_rating could be maintained as a predictor, but the remainder did not add value to this analysis. My recommendation to the organization would be to build another logistic regression model that includes school_rating and folds in other observations in the dataset for analysis. These would consist of renovation value, square footage, number of rooms, number of windows, and price. Ideally, these values will provide a more significant explanation of variance in the model with a higher pseudo R^2, and higher p-values to explain statistical significance. Including these variables should provide a better opportunity to explain the data and solve which factors contribute to classifying a house as "Luxury," which is the general goal of the analysis.

References

OpenAI. (2025, January 26). *Response on K-Fold Cross-Validation and Model Evaluation*. ChatGPT. https://chat.openai.com/