**Task 2: Data Analysis Report**

Joseph Emmanuel Cayetano

Western Governors University

D606: Data Science Capstone Task 2

April 26, 2025

**A. Summarize the real-data research question you identified in Task 1. Your summary should include justification for the research question you identified in Task 1, a description of the context in which the research question exists, and a discussion of your hypothesis.**

The research question I identified in Task 1 is: To what extent do housing features and neighborhood characteristics influence whether a home is classified as luxury, based on a Random Forest model?

This research question is important in real estate because accurately identifying luxury homes can help with pricing, marketing, and investment decisions. Real estate agencies, developers, and listing platforms often need a fast and consistent way to classify homes. Using machine learning to automate this process can save time and make it more reliable. A model that classifies homes as luxury based on available data could reduce manual work and help people in the real estate industry make better decisions using data.

I chose this research question because it focuses on a real-world problem in the real estate industry, and it is a good opportunity to apply machine learning to a practical classification problem. Most real estate listings already include housing and neighborhood features, so it makes sense to use that data for something useful like automation.

The hypothesis for this research is:

**Null Hypothesis:** Housing features and neighborhood details do not allow the model to classify homes as luxury with an accuracy of 70% or higher (model accuracy < 70%).

**Alternate Hypothesis:** Housing features and neighborhood details allow the model to classify homes as luxury with an accuracy of 70% or higher (model accuracy ≥ 70%), and some features contribute more than others to the classification.

This hypothesis gives a clear goal for how well the model should perform. If the model reaches at least 70% accuracy, it shows that the housing and neighborhood features have enough information to help classify homes as luxury. Also, by looking at which features are most important in the model, we can see which factors matter the most. This can help people in the real estate industry make better decisions based on the data.

**B. Report on your data-collection process by describing the relevant data you collected, discussing one advantage and one disadvantage of the data-gathering methodology you used, and discussing how you overcame any challenges you encountered during the process of collecting your data.**

I collected my data by downloading a housing dataset from the course platform. The dataset contains information on 7,000 homes, with 22 columns representing different housing and neighborhood features. Each row corresponds to one home. Some of the variables include: Price, SquareFootage, NumBathrooms, NumBedrooms, BackyardSpace, AgeOfHome, RenovationQuality, CrimeRate, SchoolRating, Garage, HouseColor, and others. The target variable is IsLuxury, which indicates whether a home is classified as luxury (1) or not (0). The dataset was already structured in CSV format, which made it easy to work with.

One advantage of this data-gathering method is that the dataset was already cleaned and structured, so I didn’t need to spend time on cleaning it. This allowed me focus more on analyzing the data and building the model.

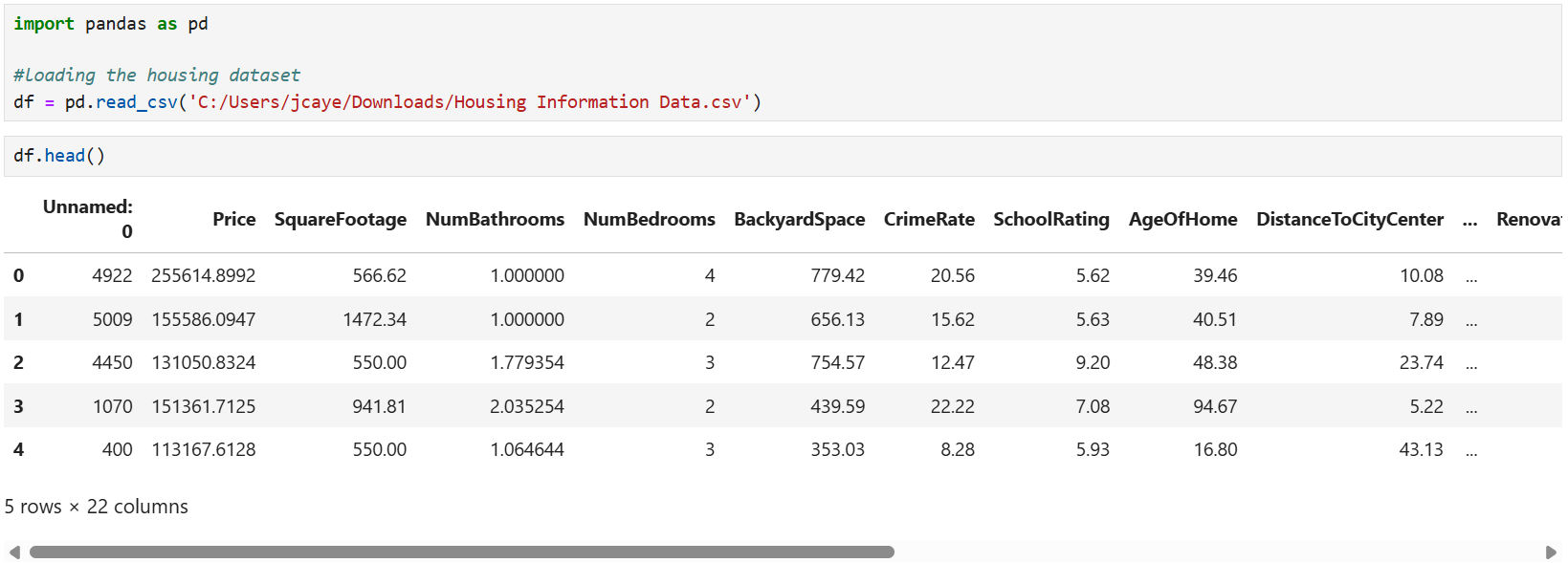
One disadvantage of this data-gathering method is that, since the data was pre-collected and provided by the course, I didn’t have control over which features were included or how they were defined. This made it harder to add useful variables or check if the data was fully accurate or realistic. Also, the dataset might not show the full complexity of real housing markets.

One challenge I encountered was making sure the housing dataset was in the right format to work with in Python. Since it was a CSV file, I needed to make sure it opened correctly and didn’t have any unexpected formatting issues. I used the read\_csv() function from the pandas library, and it loaded into my Jupyter Notebook without any problems, which allowed me to move forward with my project.

Another challenge was understanding what each column in the dataset represented and whether the column names were clear and easy to work with. To get a better idea of the data structure, I used the head() function to preview the first few rows. This helped me understand the layout and content of the dataset before starting my analysis.

**C. Describe your data-extraction and data-preparation process, and provide screenshots to illustrate each step. Explain the tools and techniques you used for data extraction and data preparation, including how these tools and techniques were used on the data. Justify why you used these particular tools and techniques, including one advantage and one disadvantage when they are used with your data-extraction and data-preparation methods.**

Data Extraction



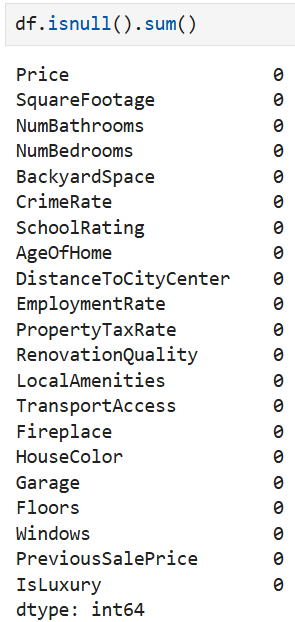
I performed data extraction by loading all columns and rows from a CSV file named “Housing Information Data.csv” into a Jupyter Notebook using the pandas library. I used the read\_csv() function to read the file and store it in a DataFrame. This allowed me to view and work with the housing dataset in a structured format. The output from the head() function confirmed that the dataset was loaded successfully.



I used the drop() function to remove the “Unnamed: 0” index column, which was automatically added during export, as it wasn’t relevant to the analysis.

Data Preparation

After the data extraction process, I performed several steps to prepare the data for analysis:



I used the isnull() and sum() functions to check for missing values in each column of the housing dataset. Since there weren’t any, I went straight to encoding the categorical variables.



I used LabelEncoder() from scikit-learn to convert categorical variables like Garage, Fireplace, and HouseColor into numeric format, since Random Forest models only work with numerical input. I used the head() function to check that the encoding worked correctly.



I separated the features and target variable by defining X as all columns except IsLuxury, and y as the IsLuxury column. I used the train\_test\_split() function from scikit-learn to first split the data into training and test sets. Then, I split the training set again to create a validation set. In the end, 60% of the data was used for training, 20% for validation, and 20% for testing.

Tools and Techniques

I used several tools and techniques for data extraction and preparation. First, I used the pandas library to load, clean, and work with the housing dataset. Specifically, I used the read\_csv() function to load the dataset into a DataFrame for analysis, and the head() function to check the structure and make sure the data loaded correctly. I used the drop() function to remove the “Unnamed: 0” index column, which wasn’t needed for the analysis, and the isnull() function to check for missing values, but there were none.

Second, I used the scikit-learn library for data preprocessing and splitting. I used the LabelEncoder() function to convert categorical variables like Garage, Fireplace, and HouseColor into numerical format, since Random Forest models require numerical input. I also used the train\_test\_split() function to divide the data into training, validation, and test sets (60%, 20%, 20%).

Finally, I used Jupyter Notebook to write and test my Python code in an interactive environment.

Justification for using these tools and techniques

I used the pandas library because it “makes it easy to work with structured data like CSV files” (Novriansyah, 2024, para. 1). It provides simple functions that make it easy to load, preview, clean, and prepare the dataset. These steps are important before starting the classification analysis. I used the read\_csv() function because it allows for quick and reliable loading of data from CSV files. I used the head() function because it helps confirm that the data loaded correctly and gives a quick view of the structure, values, and any formatting issues. I used the drop() function because it’s useful for removing columns that aren’t needed for analysis. I used the isnull() function because checking for missing values is an important part of data preparation, as missing values can affect the model’s accuracy.

I used the scikit-learn library because it is widely used in machine learning and offers tools for preprocessing and splitting data. I used the LabelEncoder() function to convert categorical variables like Garage, Fireplace, and HouseColor into numerical format, since Random Forest models require numerical input. I used the train\_test\_split() function to divide the dataset into training, validation, and test sets because this helps ensure that the model is properly trained and evaluated.

Finally, I used Jupyter Notebook because it allows me write, test, and explain my code step by step in one place. It made the data analysis process easier to follow and manage.

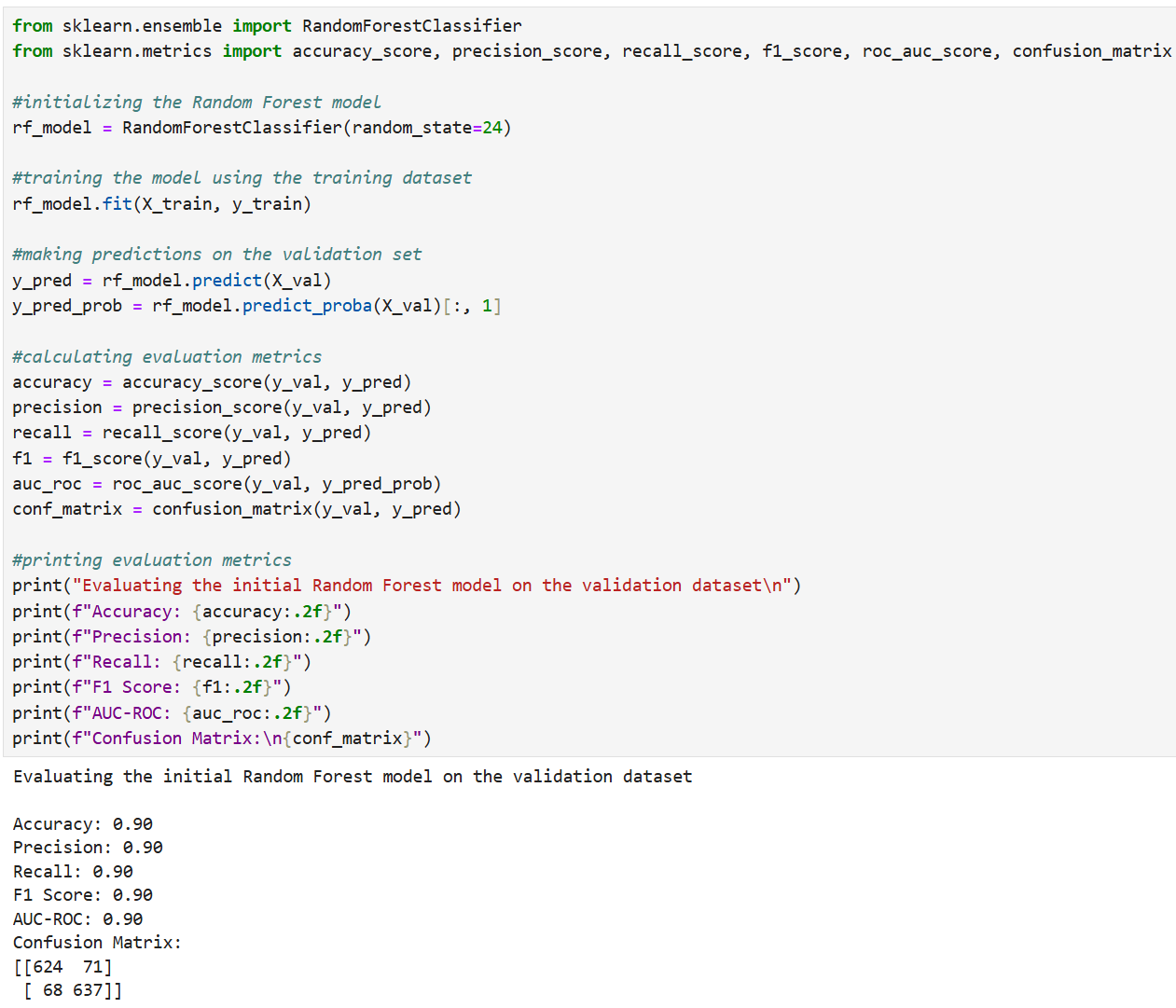
One advantage and one disadvantage of using these tools and techniques

One advantage of using libraries like pandas and scikit-learn, along with the Jupyter Notebook environment, is that they are easy to use and widely used in data science. Pandas and scikit-learn have built-in functions that make tasks like cleaning, transforming, and preparing data much easier, while Jupyter Notebook allows users like me to write and test code step by step, making the process clearer and easier to follow.

One disadvantage is that the LabelEncoder() function assigns numeric values based on alphabetical order, which can unintentionally suggest a ranking between categories. This can be confusing for nominal variables like HouseColor, although it usually doesn’t affect tree-based models like Random Forest. Also, while Jupyter Notebooks are easy to use, they can become messy or hard to navigate if not well organized.

**D. Report on your data analysis process by describing the analysis technique(s) you used to appropriately analyze the data. Include the calculations you performed and their outputs. Justify how you selected the analysis technique(s) you used, including one advantage and one disadvantage of each technique.**

For this analysis, I used the Random Forest classification technique to predict whether a home is classified as luxury or not based on its housing and neighborhood features. Random Forest is a supervised machine learning algorithm that creates multiple decision trees and combines their results to make more accurate predictions.



Here, I trained the initial Random Forest model on the training set and evaluated its performance using the validation dataset. I started by importing the RandomForestClassifier() function from the scikit-learn library to build and train model. I also imported evaluation functions like accuracy\_score(), precision\_score(), recall\_score(), f1\_score(), roc\_auc\_score(), and confusion\_matrix() to measure how well the model performed. After initializing the model, I trained it on the training data using the fit() function. Then, I used the predict() function to generate class predictions for the validation set and the predict\_proba() function to get class probabilities, which were needed to calculate the AUC-ROC score.

The model achieved 90% accuracy, meaning it correctly predicted whether a home was luxury in 90% of the cases. It also reached 90% precision, indicating that when the model predicted a home was luxury, it was correct 90% of the time. The recall score was also 90%, showing that the model correctly identified 90% of actual luxury homes. With an F1 score of 90%, the model demonstrated a strong balance between precision and recall. The AUC-ROC score of 90% shows the model was effective at distinguishing between luxury and non-luxury homes across different classification thresholds. The confusion matrix showed 624 true negatives, 71 false positives, 68 false negatives, and 637 true positives, indicating that the model made relatively few errors. Overall, the initial Random Forest model demonstrated strong, consistent performance with no signs of overfitting or class imbalance.

Justification for using Random Forest

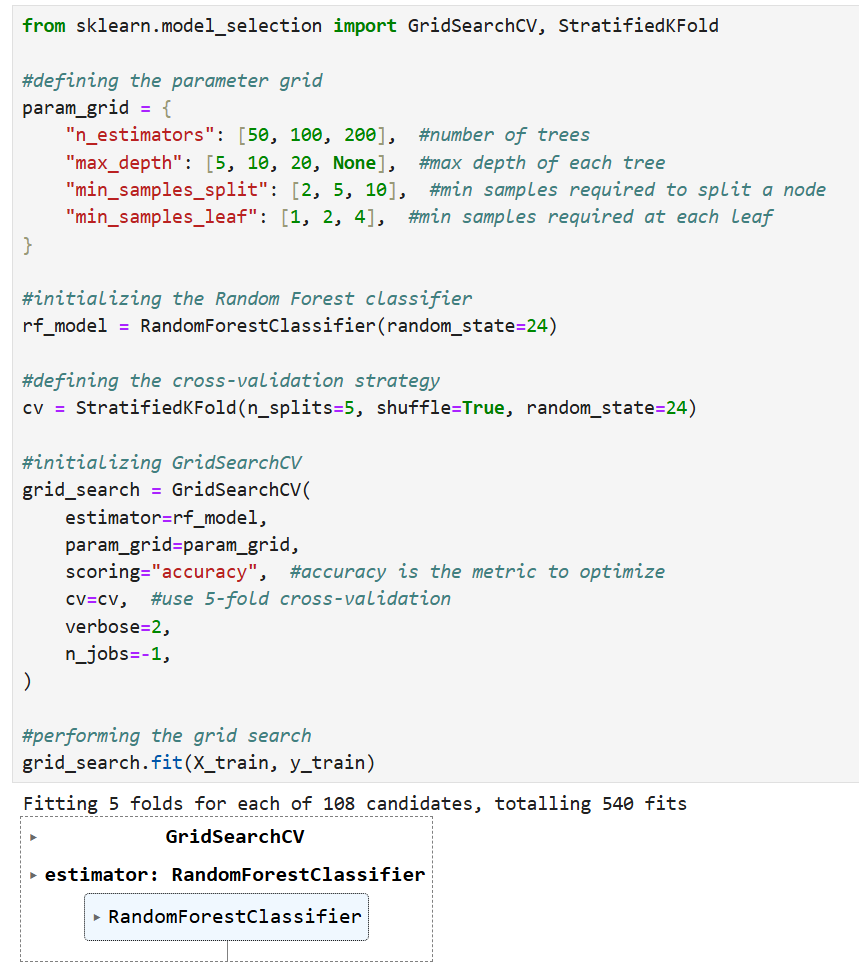
I chose Random Forest because it is reliable, works well with both numerical and categorical data, and performs well with structured datasets like the housing dataset in this project. It is also less likely to overfit compared to individual decision trees and provides feature importance scores, which shows “how much each feature contributes to the model prediction” (Zvornicanin, 2025, para. 5).

One advantage and one disadvantage of Random Forest

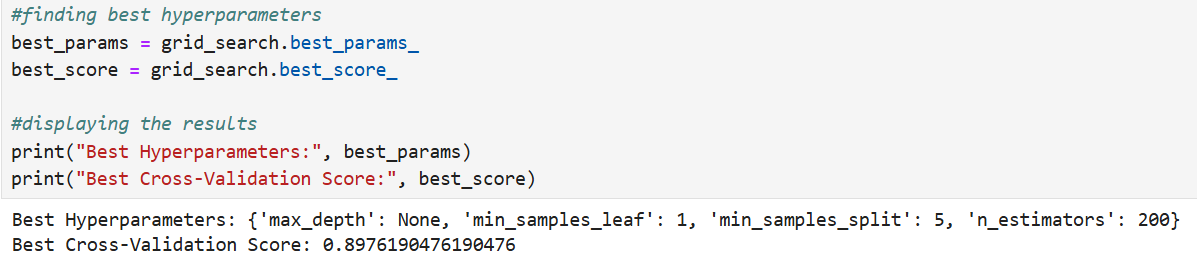
One advantage is that Random Forest is less likely to overfit compared to individual decision trees and works well with both categorical and numerical data. It also helps identify important features.

One disadvantage is that Random Forest models can be harder to interpret than simpler models like logistic regression, and training can take longer on larger datasets.

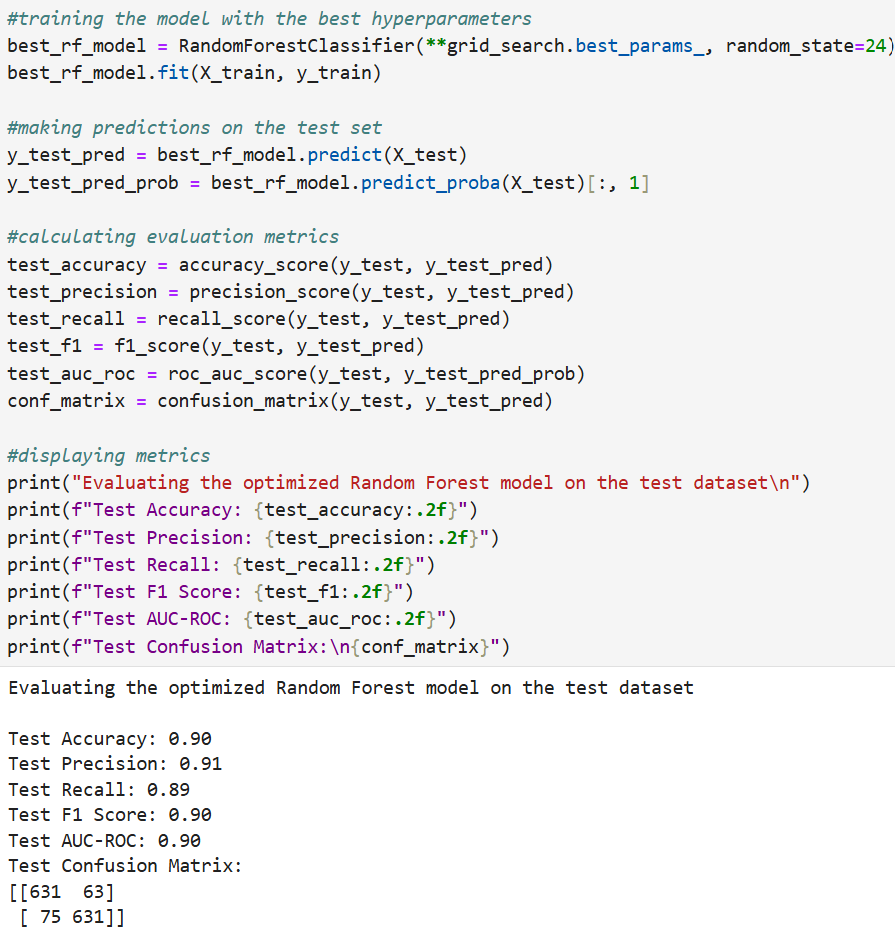
I also used hyperparameter tuning with GridSearchCV and 5-fold cross-validation to improve the model’s performance, This technique tested different combinations of hyperparameters and selected the one that gave the highest average accuracy across the validation folds.



The hyperparameters I chose to tune were ‘n\_estimators’, ‘max\_depth’, ‘min\_samples\_split’, and ‘min\_samples\_leaf’. The ‘n\_estimators’ parameter sets the number of trees in the forest. Using more trees can improve accuracy but may also increase training time. The ‘max\_depth’ parameter limits how deep each tree can grow, which helps prevent overfitting by avoiding overly complex trees. The ‘min\_samples\_split’ parameter defines the minimum number of samples needed to split a node, making the model more generalizable. The ‘min\_samples\_leaf’ parameter sets the minimum number of samples required in a leaf node to ensure that leaves are not too specific to the training data. I chose these hyperparameters because they control the model’s complexity and help balance accuracy and overfitting.



After tuning the hyperparameters using GridSearchCV with 5-fold cross-validation, I found the best combination for the Random Forest model: ‘n\_estimators = 200’, ‘max\_depth = None’, ‘min\_samples\_split = 5’, and ‘min\_samples\_leaf = 1’. This means the best-performing model used 200 trees with no depth limit, and each split required at least 5 samples, while each leaf needed at least 1 sample. These hyperparameters helped the model achieve an average accuracy of 89.76%, showing strong and consistent performance across the validation folds.



I retrained the Random Forest model using the best hyperparameters from GridSearchCV, then evaluated it on the test dataset to measure its final performance and see how well it generalized to unseen data.

The optimized model achieved 90% accuracy, meaning it correctly predicted whether a home was luxury in 90% of the cases. It also reached 91% precision, indicating that when the model predicted a home was luxury, it was correct 91% of the time. The recall score was 89%, showing that the model correctly identified 89% of actual luxury homes. With an F1 score of 90%, the model demonstrated a strong balance between precision and recall. The AUC-ROC score of 90% shows the model was effective at distinguishing between luxury and non-luxury homes across different classification thresholds. The confusion matrix showed 631 true negatives, 63 false positives, 75 false negatives, and 631 true positives, indicating that the model made relatively few errors.

The optimized Random Forest model performed very well on the unseen test set and kept high scores across all metrics. The increase in precision (from 90% to 91%) and slight drop in recall (from 90% to 89%) suggest that the model became slightly more conservative in predicting luxury homes but made fewer false positives. Overall, these results confirm that hyperparameter tuning was effective and that the model generalizes well with no signs of overfitting.

Justification for using hyperparameter tuning with GridSearchCV and 5-fold cross-validation

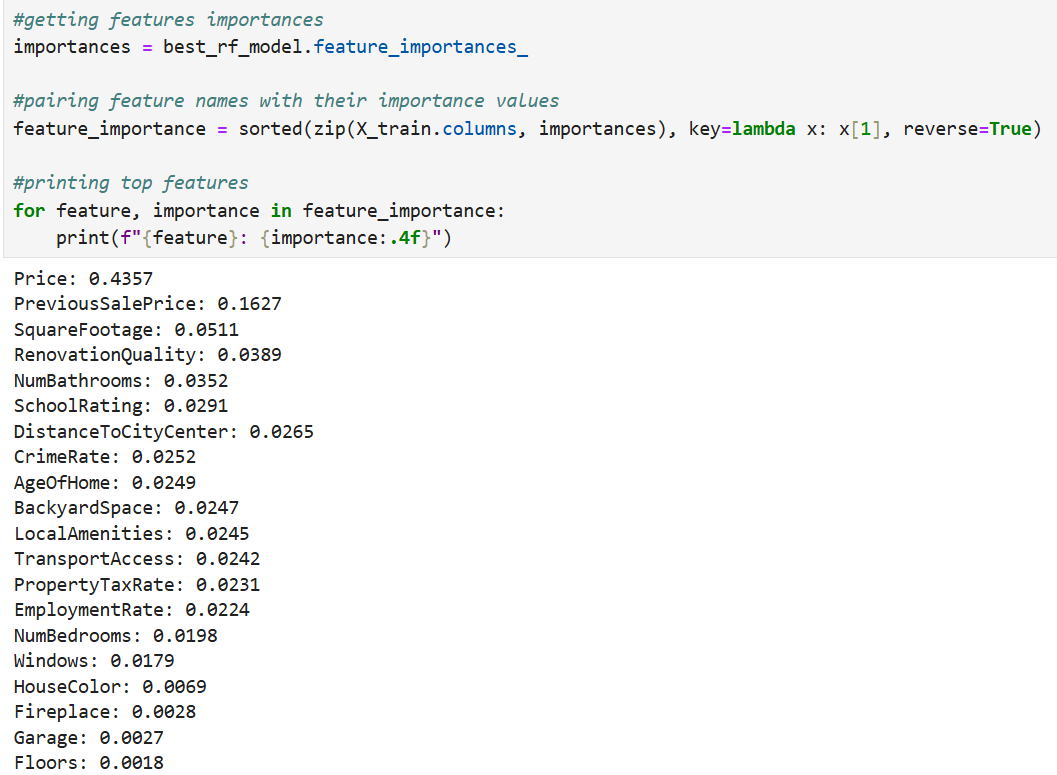
I used hyperparameter tuning with GridSearchCV and 5-fold cross-validation to improve the performance of the Random Forest model by finding the best combination of parameters. GridSearchCV made it easy to test different combinations of hyperparameters, and 5-fold cross-validation made sure the model was tested fairly on different parts of the training data. This technique helps prevent overfitting by averaging results across folds and makes the final model more reliable on new data. It also avoids the risk of relying on just one train-test split, which may not show the model’s true performance.

One advantage and one disadvantage of using hyperparameter tuning with GridSearchCV and 5-fold cross-validation

One advantage of using GridSearchCV with 5-fold cross-validation is that it reliably finds the best combination of hyperparameters. It tests the model on different parts of the training data, which helps prevent overfitting and gives a better estimate of how the model will perform on new data.

One disadvantage of using GridSearchCV with 5-fold cross-validation is that it can take a lot of time, especially when testing many hyperparameter combinations or working with large datasets, because the model has to be trained and validated multiple times.

I also used the feature importance (feature\_importances\_) attribute from the trained Random Forest model to identify which housing and neighborhood features had the most influence on predicting whether a home is luxury or not. This technique gives each feature a score based on how helpful it was in improving the model’s decisions during tree splits. Features with higher scores had a bigger influence on the model’s predictions.



Price is the most important feature, with a score of 0.4357, suggesting that luxury homes usually have much higher prices. The second most important feature is PreviousSalePrice, with a score of 0.1627, meaning homes with a history of high sale prices are more likely to be classified as luxury, even if their current value has changed. SquareFootage ranks third with a score of 0.0511, showing that larger homes are more likely to be seen as luxury. RenovationQuality ranks fourth with a score of 0.0389, showing that homes with high-quality renovations are more likely to be luxury. On the other hand, features like Fireplace, Garage, and Floors have very low importance scores, indicating they have little effect on the model's predictions.

Justification for using feature importance

I used the feature importance technique to understand which features had the most influence on predicting whether a home is luxury. This supports the second part of my research question, which focuses on identifying the key variables in the classification. These insights can also help real estate professionals make more informed decisions.

One advantage and one disadvantage of using feature importance

One advantage of using the feature importance technique is that it makes the model easier to understand by showing which features influence the predictions. This helps with decision-making and highlights important factors for further analysis.

One disadvantage of using the feature importance technique is that feature importance scores can be biased toward features with more categories or higher variance. They also don’t show how features interact with each other or whether their effect is positive or negative.

**E. Summarize the implications of your data analysis by discussing the results of your data analysis in the context of the research question, including one limitation of your analysis. Within the context of your research question, recommend a course of action based on your results. Then propose two directions or approaches for future study of the dataset.**

The results of my data analysis show that housing features and neighborhood characteristics strongly influence whether a home is classified as luxury. The Random Forest model achieved 90% accuracy and performed well across all classification metrics. Feature importance scores showed that price, previous sale price, square footage, and renovation quality were the most influential factors. This means that the data analysis answered the research question by showing that a machine learning model like Random Forest can accurately classify homes as luxury or not and identify which features matter most.

Additionally, since the model achieved an accuracy greater than 70%, the alternate hypothesis is accepted. This confirms that housing features and neighborhood characteristics provide enough information for the model to classify a home as luxury with strong accuracy, and that some features contribute more significantly to the classification.

One limitation of the analysis is that the dataset did not include subjective features like interior design quality, views, or luxury amenities, which can also play an important role in deciding whether a home is considered luxury. Without these features, the model might not capture everything that influences how luxury homes are classified in real-world situations.

Based on the results, I recommend that real estate agencies use the Random Forest model to automatically classify homes as luxury or not, based on available data. This would save time by reducing manual labeling and help them make faster, data-driven decisions for marketing, pricing, and listing homes. Builders and developers can also use the feature importance results to focus on features that make homes more appealing to luxury buyers, like adding more square footage or improving renovation quality.

For future study, I suggest adding more subjective features to the dataset, like interior design quality, views, or luxury amenities, to see if they improve the model’s accuracy and give better insights into what makes a home luxury. Another approach would be to test the model on datasets from different cities to see if it generalizes well and to understand how the definition of luxury changes across regions.

**References**

Novriansyah, N. (2024, September 1). *Introduction to Pandas for Data Analysis.* Medium. <https://medium.com/artificial-intelligence-101/introduction-to-pandas-for-data-analysis-59919a7a078e>

Zvornicanin, E. (2025, February 28). *What is Feature Importance in Machine Learning?.* Baeldung on Computer Science. https://www.baeldung.com/cs/ml-feature-importance