San Francisco Rent Analysis and Predictions

A Project

Presented to

Dr. Jessica Huynh-Westfall

San Jose State University

In Partial Fulfillment

of the Requirements of the Class

Spring 2025: CS 133 (Introduction to Data Visualization)

Charlene Khun, Helena Thiessen, Benny Chen, and Rongjie Mai

May 2nd, 2025

Table of Contents

Abstract

We aim to analyze San Francisco's rental market trends using data from the Rent Board Housing Inventory. Our project involves data cleaning, exploratory data analysis, and the development of machine learning models to predict rental prices. It has highlighted the key factors influencing rental prices and offering insights for renters, policymakers, and investors to make informed decisions in San Francisco's competitive housing market. We will navigate housing challenges, promote affordability, and contribute to a more balanced rental landscape by understanding these trends.

Background

As we know, San Francisco's housing market has a high cost of living. It is important to understand the factors that influence rental trends in this city for a variety of stakeholders, including residents, policymakers, and investors. The city's unique geographical constraints, coupled with a booming tech industry and limited housing supply, have contributed to a competitive rental market with significant fluctuations in prices over time. Renters in San Francisco are facing challenges due to the city's high cost of living. Affordable housing is scarce, and competition for available units is fierce. The ability to predict and understand rent fluctuations is vital for individuals and families seeking housing in San Francisco. This information empowers renters to make informed decisions about their housing choices, negotiate effectively, and potentially mitigate the impact of rising rent prices on their budgets.

Understanding rental trends is crucial for policymakers and urban planners in formulating strategies to address housing affordability and promote equitable access to housing for all residents. By identifying areas with high rental costs and understanding the factors driving these trends, policymakers can design targeted interventions. For example, implementing rent control or developing affordable housing to ease the burden on renters and create a more balanced housing market.

Our group's analysis utilizes data from the Rent Board Housing Inventory, a comprehensive dataset provided by the San Francisco Rent Board. The dataset captures information about rental units across the city, including rent prices, unit characteristics, and location details. By leveraging this rich data source, we want to use this project to shed light on the intricate dynamics of the San Francisco rental market as well as provide valuable insights into rental trends and their implications for various stakeholders.

In reference to the San Francisco rent dataset, we formulated five unique questions to explore and answer based on the data. For each question, we created a visual representation to strengthen our analysis. The questions we are solving are as followed:

1. How does location contribute to the overall renting price?
2. Which neighborhood has the most amount of old apartment buildings built in 1886 and before?
3. How does the number of bedrooms affect the monthly rent?
4. Which factor has the biggest impact on the monthly rent?
5. How does the average rent and other price-affecting elements differ among neighborhoods?

These questions were created to help us gain a better understanding of how each factor contributes to the rent price. Before we began our analysis, we believed that rent was generally influenced by three factors: the number of bedrooms, the year the property was built, and location.

Organization and Design of Program

Data Cleaning and Preprocessing

To begin, it is necessary to install and import the required libraries including Pandas, Seaborn, Matplotlib, and all others mentioned in the source code. Then, utilizing Pandas features, the CSV dataset can be loaded into a data frame from the supplied URL. Once the data is in a data frame format, df.info() and df.describe() are used to identify the types of data as well as any inconsistencies in the data. Due to the large volume of data, missing values were handled through dropping any rows that had a missing value for the important features. Additionally, any columns that were unnecessary for analysis were dropped to simplify the dataframe. Columns that were not the desired type were corrected and standardized by creating maps to convert object columns into numerical or Boolean types. Finally, new columns were added to engineer new features like “Cost per SqFt” from existing features.

Exploratory Data Analysis (EDA)

Different types of plots were used to make determinations about the dataset and decide what features should be included in the machine learning model. The plots that we used for exploratory analysis were histograms, correlation matrices, choropleth/folium mapping, and pairwise plots.

- Histograms allowed for visualizing distributions of key features. Based on the results, outlier values that may affect price trends can be removed.
- A correlation matrix explores correlations between numerical features to identify which columns affect price.
- After converting the data from wide to long to isolate the neighborhoods, an interactive choropleth map is generated to analyze regional rent patterns and determine if neighborhood has an impact on rent prices.
- Pairwise graphs of numerical columns are utilized to visualize linear relationships between features.

Explore and Answer Five Unique Questions

1. **How does location contribute to the overall renting price?** Supervisor districts 3 and 5 are the most affordable on average, while districts 2, 6, and 10 are the most expensive. Chinatown, Nob Hill, and Downtown/Civic Centre see most of their rent distribution on the lowest end of the range. Russian Hill, South of Market, and Bernal Heights see a very even distribution of all price ranges. Marina, Potrero Hill, Financial District, and Golden Gate Park areas have very little distribution on the affordable end of the rent scale.
2. **Which neighborhood has the most apartment buildings built in 1886 and before?** Western Addition and Mission have the most buildings built before 1887.

3. **How does the number of bedrooms affect the monthly rent?** As the bedroom count increases, the average monthly rent increases. There is a larger increase from one bedroom to two bedrooms, but a smaller increase from two bedrooms to three.
4. **Which factor has the biggest impact on the monthly rent?** Square footage has the largest positive impact on monthly rent, while “includes natural gas” and “includes refuse recycling” has the largest negative impact on monthly rent.
5. **How does the average rent and other price-affecting elements differ among neighborhoods?** The Financial District, Potrero Hill, and Glen Park have the highest average rents of all neighborhoods. Among those, the Financial District has the smallest average square footage, making it the worst value neighborhood. Chinatown and Downtown/Civic Center have the lowest average rents but also a very low average square footage, making them a compromise not suited to all renters. The next lowest rent regions are Outer Richmond, Visitacion Valley, and Bayview. Of these, Bayview has the lowest average rent and the highest average square footage, making it the best value.

Machine Learning Model Selection and Training

We first split the dataset into training and testing sets. To ensure an accurate distribution of build years, we applied stratified sampling. Categorical columns were encoded using one-hot encoding. All data manipulations were performed through a pipeline to maintain consistency and reproducibility. We then trained three regression models: Linear Regression, Decision Tree Regression, and Random Forest Regression.

The reason we chose these three different models is because they each offer unique strengths and allow for a well-rounded comparison. Linear Regression provided us with a simple way to interpret our data and assumes a linear relationship between the features and rent. Decision Tree Regression captures non-linear patterns and interactions without requiring feature scaling. Random Forest Regression improves accuracy and reduces overfitting by combining multiple decision trees.

Finally, we evaluated each model's performance using metrics such as Root Mean Squared Error (RMSE).

Model Optimization

We used cross-validation scoring based on RMSE, MAE, and the R^2 value to compare the performance of the models. On the training data, Random Forest Regression had the best score for all key metrics (RMSE, MAE and R^2 value), so it was selected as the best-performing model to optimize further. We then fine-tuned its hyperparameters using GridSearchCV to improve its accuracy. Finally, we analyzed the feature importances to identify the key factors influencing rent predictions.

Model Evaluation and Testing

We evaluated the final model on the testing set to assess its ability to generalize new data. To compare predicted rent values with actual rent values, we created a scatter plot. Additionally, we analyzed the model's predictions using confusion matrices to assess its accuracy and identify any patterns of misclassification.

Project Results

Graphs for Initial Data Exploration

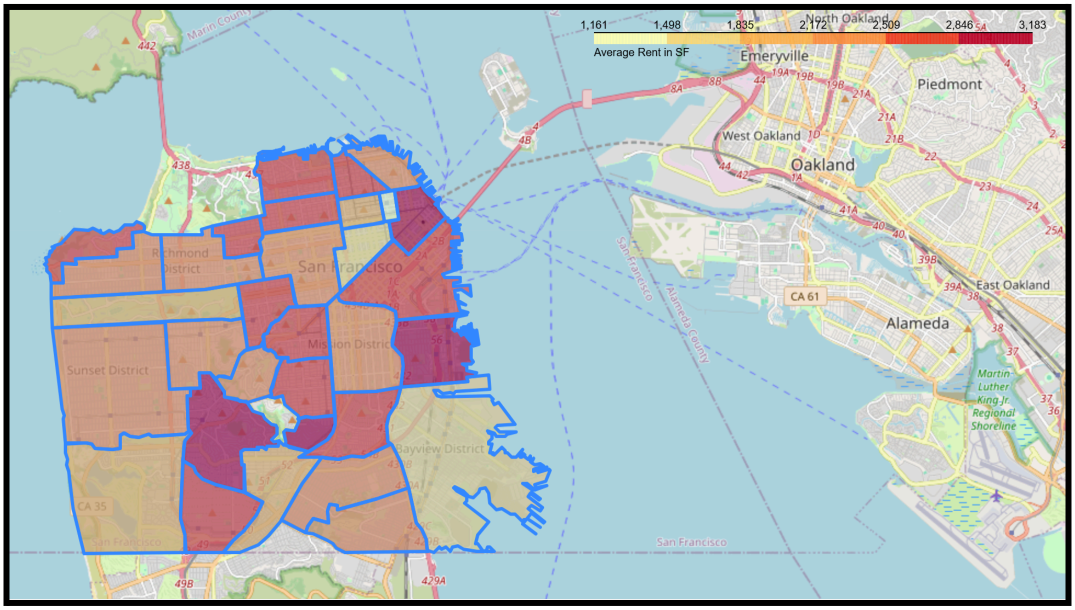


Figure 1 - Map displaying the average rent of different neighborhoods in San Francisco.

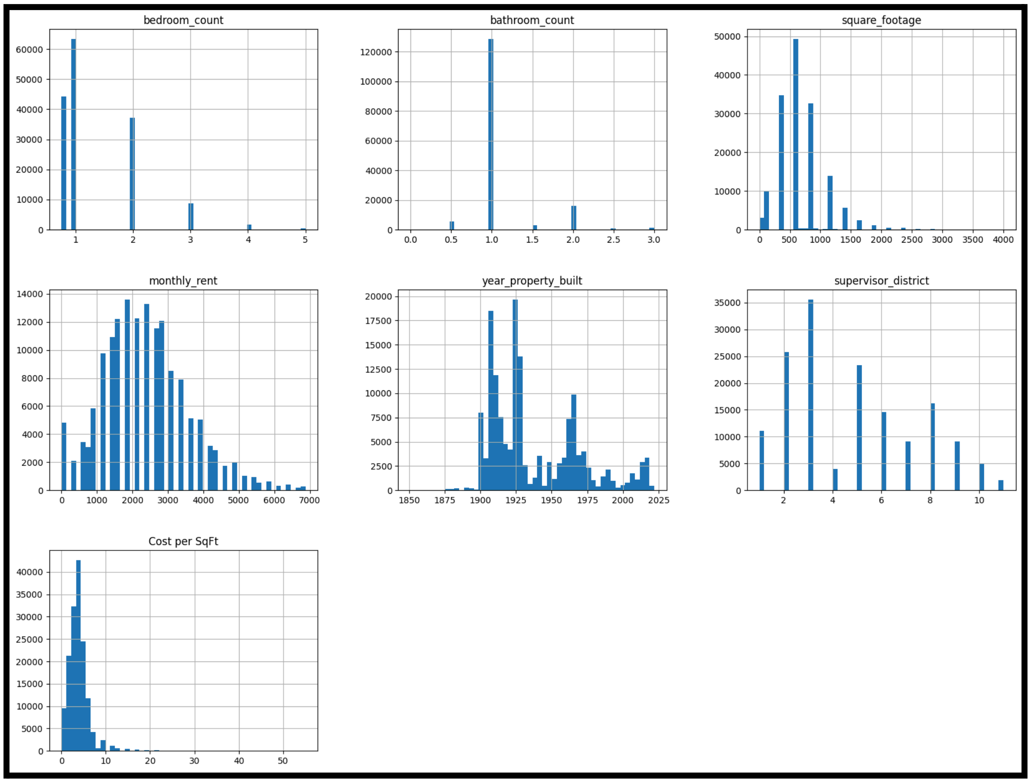


Figure 2 - A histogram displaying the average values of each rent feature (helps determine outliers).

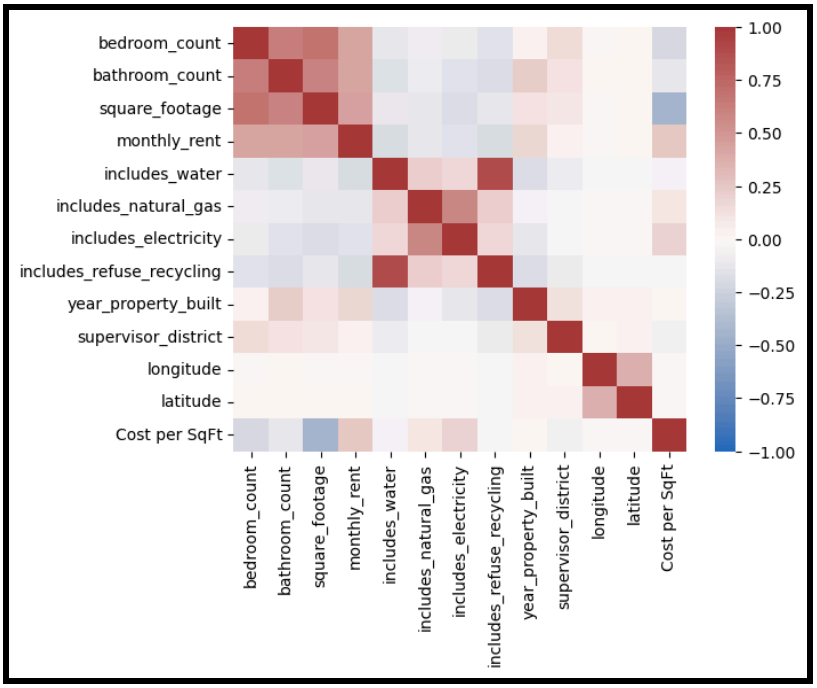


Figure 3 - A correlation matrix that illustrates how rent features are correlated with each other. Strong correlations with monthly_rent tell us what drives price and are useful for modeling or market insights.

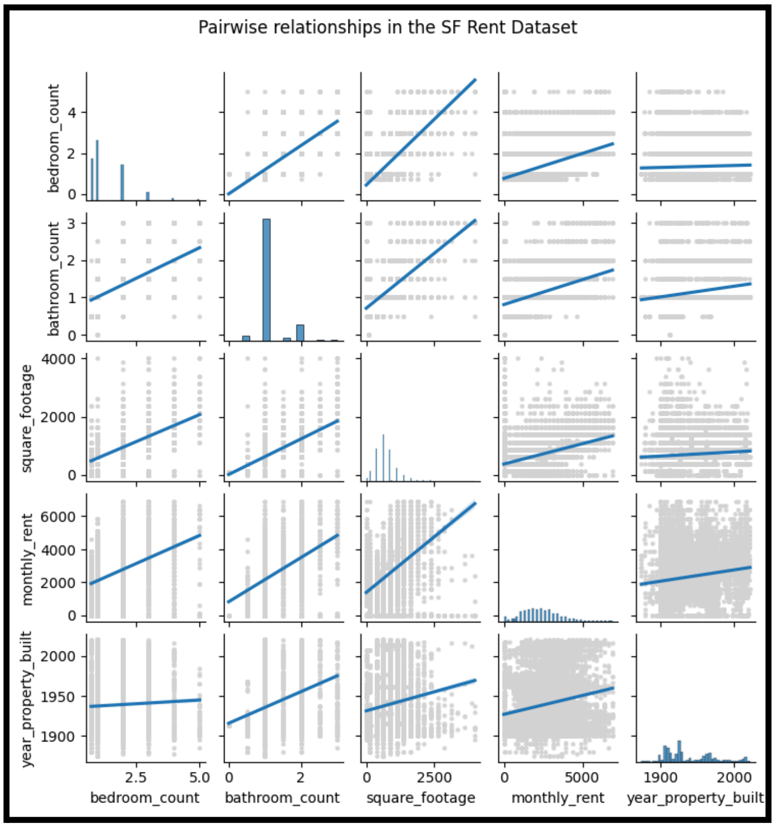


Figure 4 - A pairplot that illustrates the linear pairwise relationships for key numerical rent features.

Data Exploration Results

Exploratory graphing revealed which factors influenced rental costs. This helps identify which features should be included in a machine learning model to optimize price prediction. The choropleth/folium mapping illustrates that neighborhood, a categorical value, has an impact on rent price. The histogram provided a means to identify outliers that were then dropped to optimize the data. Lastly, the numerical values were assessed through pairwise plotting and a correlation matrix, which showed that rent increases with greater square footage, bedroom count, bathroom count, and building year. We noticed rent prices were lower when utilities, waste, electric, and gas were included.

Description of test data and test strategy

We selected monthly_rent as the target variable for our analysis. The dataset was preprocessed by applying one-hot encoding to the categorical feature analysis_neighborhood and using median imputation and ordinal standardization for the numerical features. These steps were combined using a ColumnTransformer, and the resulting feature matrix was converted into a sparse DataFrame for efficient storage and modeling. We then applied three machine learning models to the rent_prepared pipeline: Linear Regression, Decision Tree Regression, and Random Forest Regression. The model with the lowest error score and highest R^2 value will be selected for further evaluation and analysis.

Assessing and Choosing a Model

The metrics that were chosen to assess the machine learning models were root mean squared error (RMSE), mean absolute error (MAE), and R^2. These were chosen because they assess the difference between actual and predicted values on regression models. The models were then evaluated by assessing the mean score of a ten-fold cross-validation of these metrics. Random Forest Regression performed best on all three metrics, so it was chosen as the best model to fine-tune for use on the test data. Random Forest Regression was then run through Grid Search and Random Search. The best results from all three optimization methods were then compared based on RMSE, MAE, and R^2 of the best estimator. The scores were extremely similar on training data. Although Random Search slightly outperformed Grid Search on the test data in the table below, we chose Grid Search based on its more consistent performance in prior iterations and similarly strong training scores.

Training Data Results

	Linear Regression	Decision Tree Regression	Random Forest Regression
RMSE	748.00	658.89	639.68
MAE	581.35	466.02	464.93
R^2	0.43	0.56	0.58

	Grid Search	Random Search
RMSE	541.87	541.39
MAE	394.66	394.27
R^2	0.7011	0.7016

Testing Random Forest Regression Model

Our code in the “Analyze the Best Models and Their Errors” section evaluates the performance of the Random Forest Regression model selected through grid search by extracting its feature importances to understand which variables most influence predictions. It combines numerical attributes with one-hot encoded categorical features to match each feature with its importance score, then ranks them from most to least important. Next, the code prepares the test data (using the same preprocessing pipeline), generates predictions using the final model, and calculates evaluation metrics, including RMSE, MAE, and R^2 to measure the model’s prediction accuracy on unseen data. When running on the test data, the model has a large margin of error, apparent in the RMSE, MAE, and R^2. Although there were many attempts made at fine-tuning the model to achieve a better result, the score was ultimately determined to be a consequence of the data supplied.

Test Data Results

	Random Forest Regression
RMSE	621.53
MAE	455.27
R^2	0.6115

In Figure 5 (see next page), we have a confusion matrix that evaluates how well the model classifies rental price predictions into categorical groups—Low, Medium, and High—based on their numeric values. The model correctly placed 84% of high-cost apartments, 68% of medium-cost apartments, and 24% of low-cost apartments. This model is fairly accurate at predicting high-cost apartments, but it consistently guesses too high for apartments below that threshold. The error observed in RMSE and MAE clearly trends towards predicting the rent cost as higher than it actually is. While this model is able to perform some degree of prediction, it is ultimately not an ideal model and should be revised with more supporting data that can form better predictions.

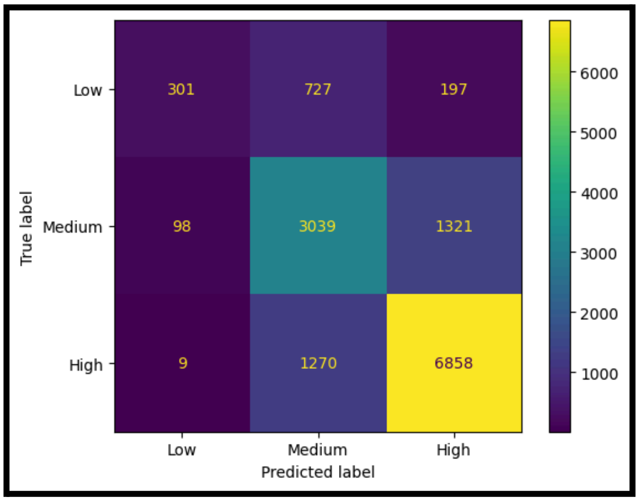
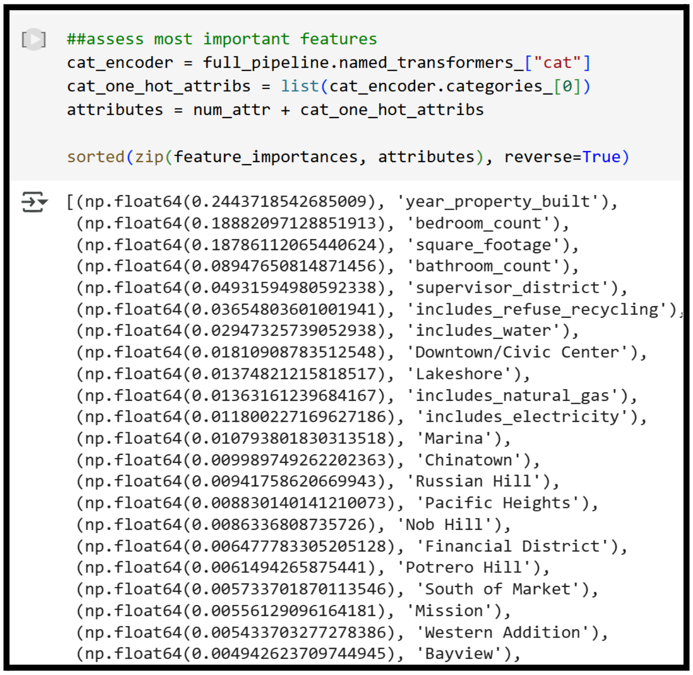


Figure 5 - Confusion matrix based on random forest regression test set.

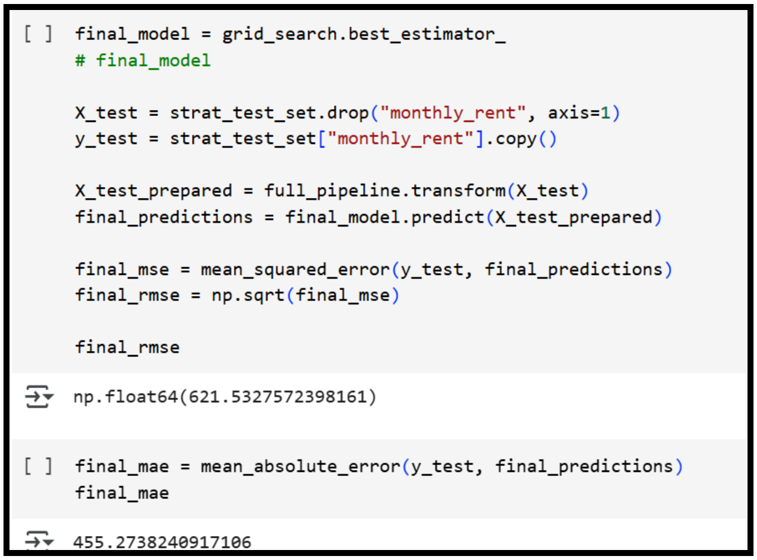
Screenshots of Important Code and Output



```
##assess most important features
cat_encoder = full_pipeline.named_transformers_["cat"]
cat_one_hot_attribs = list(cat_encoder.categories_[0])
attributes = num_attr + cat_one_hot_attribs

sorted(zip(feature_importances, attributes), reverse=True)
```

```
[(np.float64(0.2443718542685009), 'year_property_built'),
 (np.float64(0.18882097128851913), 'bedroom_count'),
 (np.float64(0.18786112065440624), 'square_footage'),
 (np.float64(0.08947650814871456), 'bathroom_count'),
 (np.float64(0.04931594980592338), 'supervisor_district'),
 (np.float64(0.03654803601001941), 'includes_refuse_recycling'),
 (np.float64(0.02947325739052938), 'includes_water'),
 (np.float64(0.01810908783512548), 'Downtown/Civic Center'),
 (np.float64(0.01374821215818517), 'Lakeshore'),
 (np.float64(0.01363161239684167), 'includes_natural_gas'),
 (np.float64(0.011800227169627186), 'includes_electricity'),
 (np.float64(0.010793801830313518), 'Marina'),
 (np.float64(0.009989749262202363), 'Chinatown'),
 (np.float64(0.00941758620669943), 'Russian Hill'),
 (np.float64(0.008830140141210073), 'Pacific Heights'),
 (np.float64(0.00863368087357 26), 'Nob Hill'),
 (np.float64(0.006477783305205128), 'Financial District'),
 (np.float64(0.0061494265875441), 'Potrero Hill'),
 (np.float64(0.005733701870113546), 'South of Market'),
 (np.float64(0.00556129096164181), 'Mission'),
 (np.float64(0.005433703277278386), 'Western Addition'),
 (np.float64(0.004942623709744945), 'Bayview'),
```

Figure 6 - Gathering all features to see which features will influence the predictions the most



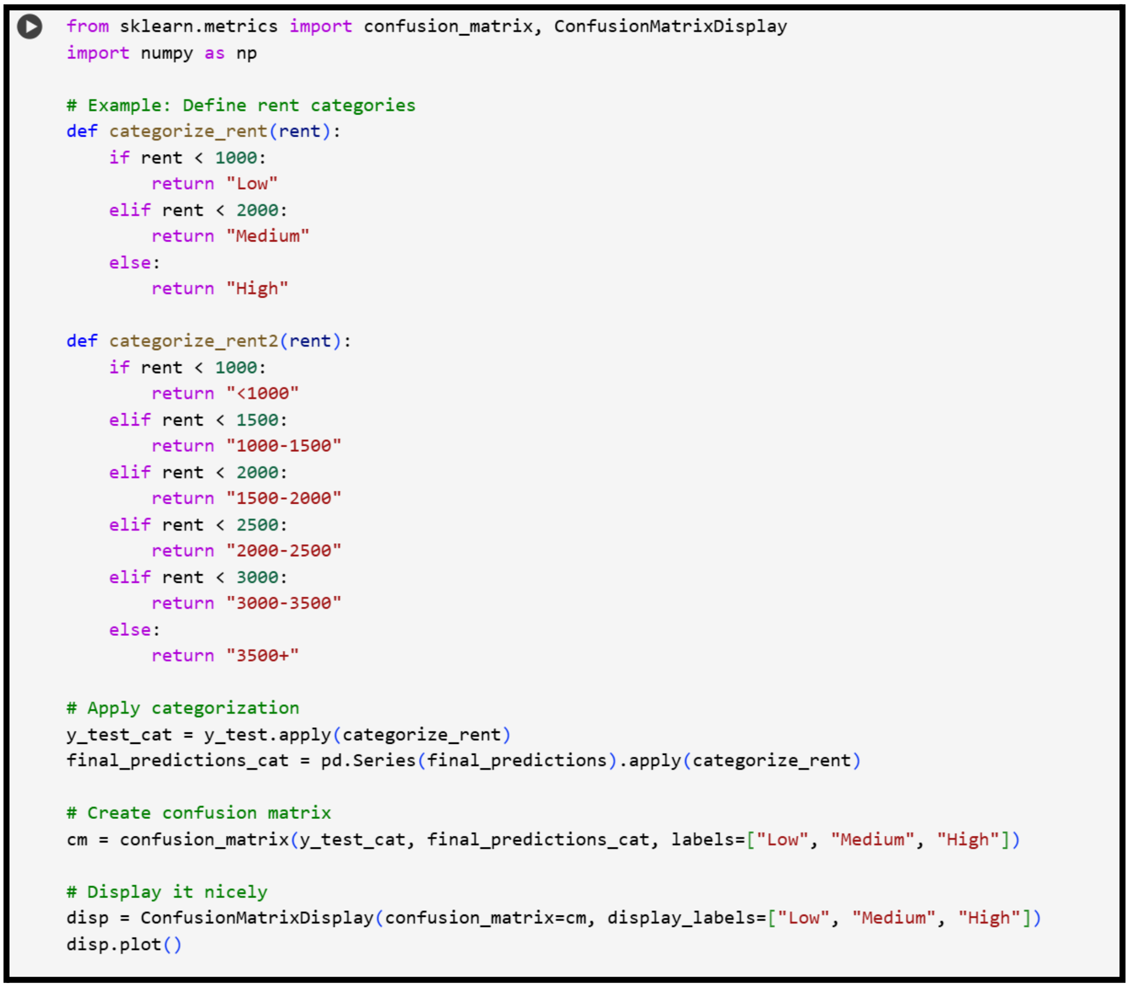
```
final_model = grid_search.best_estimator_
# final_model

X_test = strat_test_set.drop("monthly_rent", axis=1)
y_test = strat_test_set["monthly_rent"].copy()

X_test_prepared = full_pipeline.transform(X_test)
final_predictions = final_model.predict(X_test_prepared)

final_mse = mean_squared_error(y_test, final_predictions)
final_rmse = np.sqrt(final_mse)

final_rmse
```

```
np.float64(621.5327572398161)
```

```
final_mae = mean_absolute_error(y_test, final_predictions)
final_mae
```

```
455.2738240917106
```

Figure 7 - Gathering predictions, RMSE, and MAE for the final model.



```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import numpy as np

# Example: Define rent categories
def categorize_rent(rent):
    if rent < 1000:
        return "Low"
    elif rent < 2000:
        return "Medium"
    else:
        return "High"

def categorize_rent2(rent):
    if rent < 1000:
        return "<1000"
    elif rent < 1500:
        return "1000-1500"
    elif rent < 2000:
        return "1500-2000"
    elif rent < 2500:
        return "2000-2500"
    elif rent < 3000:
        return "3000-3500"
    else:
        return "3500+"

# Apply categorization
y_test_cat = y_test.apply(categorize_rent)
final_predictions_cat = pd.Series(final_predictions).apply(categorize_rent)

# Create confusion matrix
cm = confusion_matrix(y_test_cat, final_predictions_cat, labels=["Low", "Medium", "High"])

# Display it nicely
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=["Low", "Medium", "High"])
disp.plot()
```

Figure 8 - Creating a confusion matrix to compare predicted values and actual values. To see the results, see Figure 5 on page 12 of the report.

Final Remarks and Conclusion

Insights

The analysis of the San Francisco rent dataset revealed that the Random Forest Regression model with hyperparameter tuning (via Grid and Random Search) produced the most accurate predictions among all models tested. Before tuning, the Random Forest already outperformed Linear and Decision Tree Regression models, with lower RMSE (639.68) and MAE (464.93) and a higher R^2 value (0.58). After tuning, performance improved further, with Random Search achieving the best RMSE (541.39) and MAE (394.27) with an R^2 value of 0.7016. It's important to note that the scores varied with each run. We conducted several iterations, and across those runs, grid search consistently produced the best scores. For this reason, we chose to use Grid Search in our code. The final evaluation of the model on the test set yielded an RMSE of 621.53, an MAE of 455.27, and an R^2 value of 0.6155 reflecting moderate accuracy and acceptable generalization. While the model can reasonably predict rental prices, it is not highly precise, emphasizing the inherent complexity of rental pricing. These results also highlight the importance for renters to consider multiple features, such as property location, size, and included amenities, when evaluating housing options in San Francisco.

Challenges

One of the key challenges encountered was the model's slow runtime, which hindered the ability to efficiently test and refine the system. Additionally, the model exhibited a relatively large margin of error, limiting its effectiveness in making accurate predictions. These issues highlight the limitations of the current approach in handling complex, nonlinear patterns and suggest the need for more advanced techniques.

Improvements

To improve the machine learning aspect of this project, future work could explore advanced models like Gradient Boosting (e.g., XGBoost, LightGBM) or neural networks to better capture nonlinear patterns. Enhancing feature engineering with interaction terms or domain-specific variables (e.g., walkability scores) could also boost performance. Further improvements may come from collecting more detailed data about the apartments to help uncover stronger relationships with rental cost. Additionally, techniques such as SHAP values for interpretability and regularization to reduce overfitting can make the model more transparent and reliable. The consideration of all these improvements will lead to a more accurate, transparent, and robust model for predicting rental prices.

How to Run the Code

1. **Install the necessary Libraries**: Ensure you have the following Python libraries installed: pandas, seaborn, matplotlib, re, folium, geopy, scikit-learn, and scipy.
    a. You can install them using pip: pip install pandas seaborn matplotlib re folium geopy scikit-learn scipy OR !pip install pandas seaborn matplotlib folium geopy scikit-learn scipy (when using Jupyter Notebooks on Google Colab).
2. **Access the Dataset**: Obtain the Rent Board Housing Inventory dataset from the provided URL from the “References” page on page 16 of the report.

3. **Execute the Code**: Run the Python code blocks in a Jupyter Notebook (from Google Colab) or a similar environment. The code performs data cleaning, exploratory analysis, model training, and evaluation.
4. **Interpret Results**: Analyze the generated visualizations, model performance metrics, and feature importances to gain insights into San Francisco's rental market trends.

References

- The dataset used for this project was obtained from SF Rent Board Housing Inventory by DataSF: https://data.sfgov.org/Housing-and-Buildings/Rent-Board-Housing-Inventory/gdc7-dmcn/about_data

  Specifically, we decided to use the data from April 8th, 2025.

  Source: "[https://raw.githubusercontent.com/tlena43/DataVis/refs/heads/main/Rent_Board_Housing_Inventory_20250408.csv]"

- This work utilizes several Python libraries, including:

  Pandas, Seaborn, Matplotlib, Re, Folium, Geopy, SKlearn, Scipy