CMPSC 445 (Spring 2024)
Machine Learning

Machine Learning Project: Predicting House Prices in Philadelphia

Sarim Kazmi

Submitted On: March 6, 2024

TABLE OF CONTENTS

1. Objective

The primary objective of this project is to predict the market values of houses using various features from the property data in Philadelphia. We achieve this by performing a regression analysis on a dataset containing various features related to properties, aiming to predict the market values of these properties. Four different regression models: Linear Regression, Decision Tree, Random Forest, and Gradient Boosting are trained on the training set and evaluated on the testing set using metrics such as Mean Squared Error (MSE) and R-squared (R2).

2. Data Collection

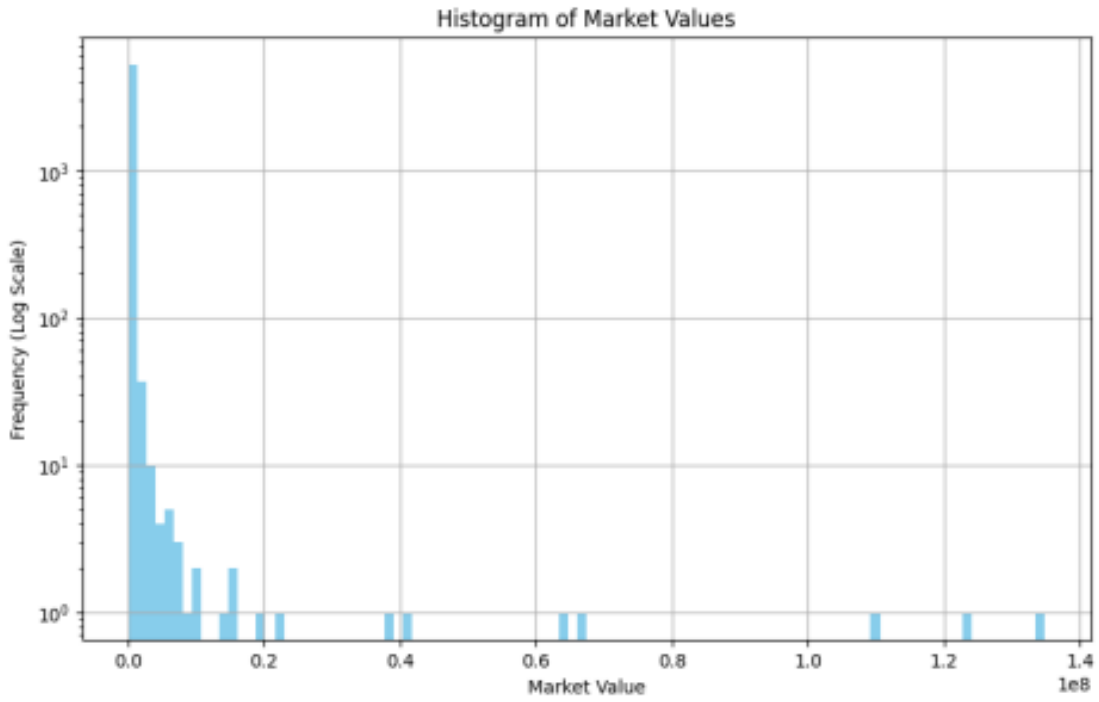
- **Dataset Source:** The data for this project was collected from the official website of the Philadelphia government.
Link: Dataset
- **Data trimming:** This data was more than 50000 so we narrowed it down to about 5200. I chose this data as it held relevant information that could help us with the project, such as variables: zip code, number of rooms, number of bathrooms, year built, etc.

3. Data Exploration

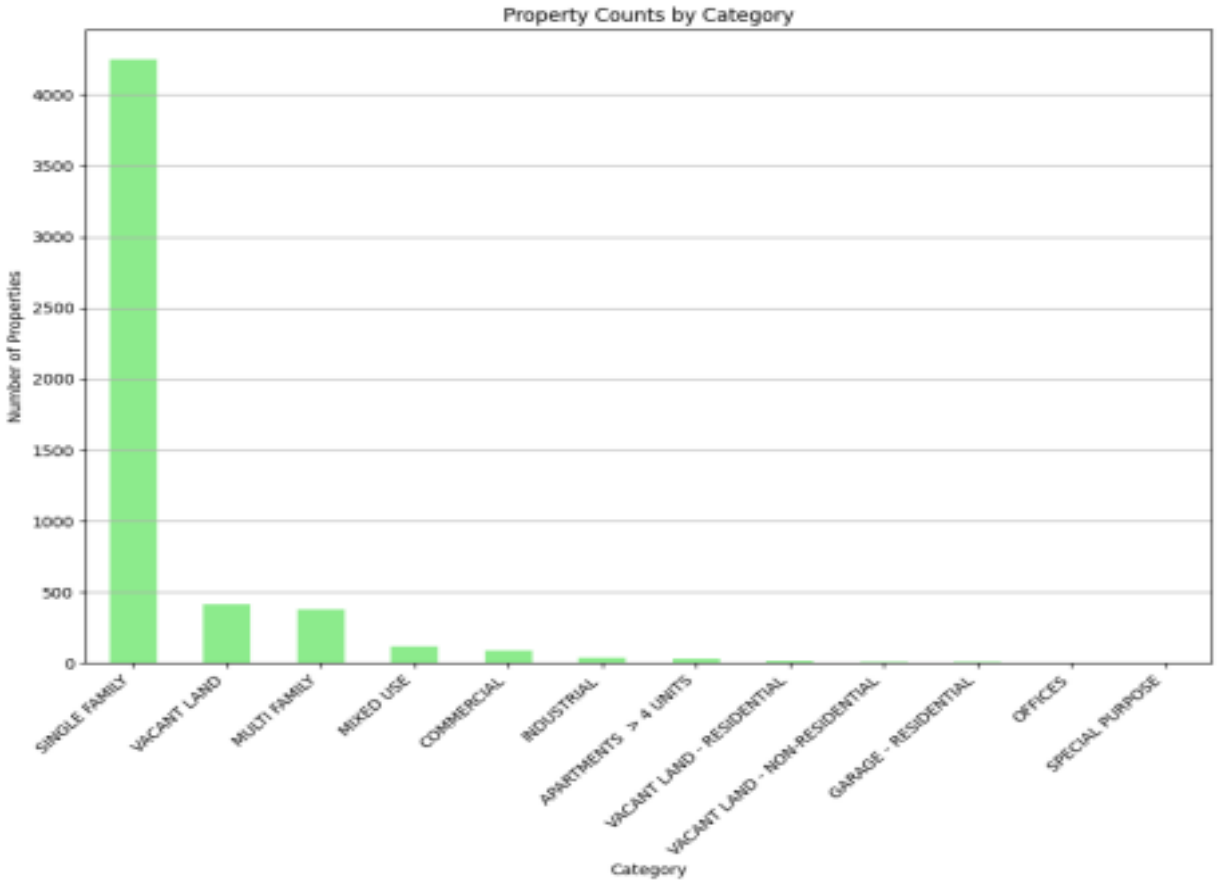
- **Data Loading:** The dataset is loaded into a pandas DataFrame from a CSV file. The initial examination with `data.head()` reveals the structure and types of data available.
- **Data Description:** Using `data.describe()`, statistical summaries of the numerical columns are obtained, providing insights into the central tendency, dispersion, and shape of the dataset's distribution.
- **Data Shape:** The shape of the dataset is explored using `data.shape`, indicating the total number of entries and features.
- **Data Information:** `data.info()` is used to obtain a concise summary of the DataFrame, including the number of non-null entries in each column, which helps in identifying missing values.

4. Data Visualization

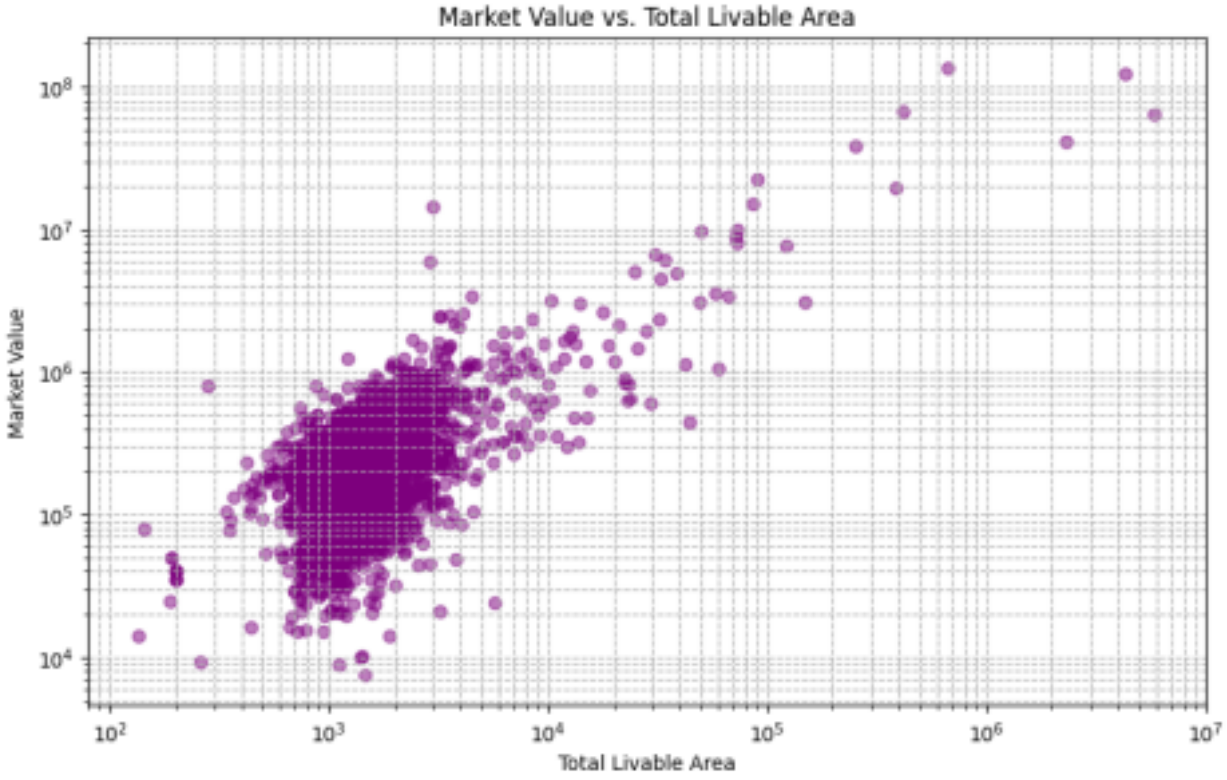
- **Histogram of Market Values:** A histogram is generated to visualize the distribution of market values. This helps in understanding the range and commonality of property values.



- Property Counts by Category: A bar chart displays the counts of properties across different categories, indicating the diversity of property types in the dataset.



- **Market Value vs. Total Livable Area:** A scatter plot explores the relationship between market value and total livable area, suggesting potential correlation between property size and its market value.



5. Data Preprocessing

- **Feature Selection:** A subset of relevant features is selected, focusing on those likely to influence the market value, such as `zip_code`, `building_code_description`, and `total_livable_area`.

- **Handling Missing Values:** Missing values are addressed through imputation, with numerical missing values filled in with the median and categorical missing values filled in with the mode.

- **Encoding Categorical Variables:** Categorical variables are transformed into a numerical format using one-hot encoding.

6. Feature Engineering

- **Meaningful Information:** We extract meaningful information from the column 'year_built' to instead show 'years_since_built' as it could help predict how old the house is more accurately.
- **Transforming Variable:** We transform the variable 'total_livable_area' to 'sqrt_total_livable_area' which is the square root of the original livable area so that the data is normalized and is scaled with

other numerical features.
- **Target Variable:** We set our target variable as 'market_value' and drop the target variable from the data so it can be predicted.

7. Construction of Test Dataset and Model Development

- **Data Splitting:** The dataset is split into training and testing sets to evaluate model performance.
- **Model Selection:** Several models are trained and evaluated, including Linear Regression, Decision Tree, Random Forest, and Gradient Boosting. The models are compared based on Mean Squared Error (MSE) and R^2 score.

8. Model Optimization

- **Hyperparameter Tuning:** GridSearchCV is employed to optimize the hyperparameters of a Random Forest model. The hyperparameters under consideration include the number of estimators (trees) in the forest (n_estimators), the maximum depth of each tree (max_depth), the minimum number of samples required to split an internal node (min_samples_split), and the minimum number of samples required to be a leaf node (min_samples_leaf). Once the grid search is complete, the best hyperparameters are identified, and a Random Forest model is trained with these optimized settings.

9. Model Evaluation

```
{'Linear Regression': {'MSE': 11281671721716.588, 'R2': 0.4041470090161613},
 'Decision Tree': {'MSE': 7603894331826.986, 'R2': 0.5983925704891213},
 'Random Forest': {'MSE': 5002845320063.591, 'R2': 0.7357696252009169},
 'Gradient Boosting': {'MSE': 8415396865208.042, 'R2': 0.5555322370533164}}
```

After running all the models, The Random Forest model demonstrates the best performance initially, with lower error compared to other models and performs the best among the experimented models with the highest R^2 Score of 0.735.

```
[CV] END max_depth=None, min_samples_leaf=2, min_samples_split=2, n_estimators=50; total time=   1.6s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=5, n_estimators=150; total time=   5.3s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=5, n_estimators=150; total time=   5.4s
[CV] END max_depth=None, min_samples_leaf=2, min_samples_split=2, n_estimators=100; total time=   3.2s
[CV] END max_depth=None, min_samples_leaf=2, min_samples_split=2, n_estimators=100; total time=   3.2s
[CV] END max_depth=None, min_samples_leaf=2, min_samples_split=2, n_estimators=100; total time=   3.1s
[CV] END max_depth=None, min_samples_leaf=1, min_samples_split=5, n_estimators=150; total time=   5.2s
...
[CV] END max_depth=20, min_samples_leaf=2, min_samples_split=5, n_estimators=100; total time=   2.0s
[CV] END max_depth=20, min_samples_leaf=2, min_samples_split=5, n_estimators=150; total time=   2.5s
[CV] END max_depth=20, min_samples_leaf=2, min_samples_split=5, n_estimators=150; total time=   2.5s
[CV] END max_depth=20, min_samples_leaf=2, min_samples_split=5, n_estimators=150; total time=   2.2s
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```
0.7823993561707148
```

The optimized model achieves an R^2 score of approximately 0.782, indicating a robust model capable of explaining about 78.2% of the variance in house prices.

10. Conclusion

The project successfully develops a predictive model for house market values using the RandomForestRegressor, with further optimization through hyperparameter tuning enhancing its performance. This model can serve as a valuable tool for estimating property values, assisting in real estate analysis, and decision-making processes.

11. Discussion

- **Data Analysis Insights:** During the data analysis phase, key insights were taken from exploring the dataset. Features such as 'total_livable_area,' 'number_of_rooms,' and 'year_built' demonstrated significant impacts on the market value of houses. The square root transformation of 'total_livable_area' was employed to eliminate the possibility of extreme values and achieve a more normalized distribution.
- **Challenges Encountered During Model Development:** Dealing with missing values in certain features was a challenge in predicting the market value. The use of imputation techniques, such as median imputation for numerical features and most frequent imputation for categorical features, were used to address this challenge. The choice of the most suitable model involved experimentation with Linear Regression, Decision Trees, Random Forests, and Gradient Boosting. Different models showed varying strengths and weaknesses, requiring a comprehensive evaluation to identify the most effective one.
- **Recommendations for Improving Predictive Performance:** Further exploration of feature engineering techniques could enhance model performance. Experimenting with different transformations, creating interaction terms, or extracting additional meaningful information from the available features might contribute to better predictions. Hyperparameter tuning can play a

crucial role in improving model performance. Continued exploration of hyperparameter space and potentially incorporating more detailed optimization techniques may lead to further enhancements. Investigating the robustness of the model to outliers and exploring techniques like robust regression or outlier detection may contribute to a better predictive model.

- **Strengths of our approach:** The approach involved thorough preprocessing, including handling missing values, transforming features, and employing one-hot encoding. This comprehensive preprocessing enhances the model's ability to handle many data types and ensures that the input is suitable for different algorithms. The use of GridSearchCV for hyperparameter tuning demonstrates a way to optimize model performance. Tuning hyperparameters ensures that the models are fine-tuned for the specific characteristics of the dataset.
- **Weaknesses of our approach:** While the project utilized some feature engineering techniques, further exploration could uncover additional insights. More extensive experimentation with interactions between variables might enhance the model's ability to capture complex relationships. The project did not address any potential outliers, as the data used was assumed to be consistent. Investigating and implementing techniques for handling outliers could enhance the model's stability and reliability.
- **Future Work:** Advanced hyperparameter optimization techniques such as Bayesian optimization, genetic algorithms, or hyperband offer more efficient and potentially more effective alternatives to grid search or random search, especially in large hyperparameter spaces. These techniques can intelligently navigate the search space to find optimal configurations faster and with fewer computational resources. By leveraging these strategies, one can not only enhance model performance but also uncover deeper insights into the data and problem at hand, leading to more robust and accurate predictive models.