

2024

Real State

Rent Price Prediction And Deployment

Prepared by:

Abhash Rai

Table of Contents

Table of Figures.....	3
Introduction.....	4
Problem Definition.....	4
Dataset Selection	4
Exploratory Data Analysis (EDA).....	5
Data Cleaning and Selection	7
Handling Missing Values.....	7
Feature Removal	7
Further Cleaning.....	7
Model Development and Evaluation	8
Feature Engineering and Preprocessing Pipeline	8
Model Selection	8
Data Splitting.....	8
Evaluation Metrics.....	8
Cross-Validation	8
Performance Comparison	9
Supervised Learning.....	9
Unsupervised Learning.....	9
Model Interpretation and Explainability.....	12
Deployment Strategy.....	14
Further Improvements	14
Handling Concept Drift.....	14
Conclusion	15

Table of Figures

Figure 1 Distribution of Bedrooms and Bathrooms in Rental Listings	5
Figure 2 Top 10 Most Expensive and Cheapest States and Cities	5
Figure 3 Geographical coordinates with rental price	6
Figure 4 Supervised Learning 6 models performance comparison	9
Figure 5 Elbow method and silhouette method with KMeans	9
Figure 6 DBSCAN clusters visualization of rental listings	10
Figure 7 PCA Components Visualization of Rental Listings with DBSCAN clusters	10
Figure 8 t-SNE Components Visualization of Rental Listings with DBSCAN clusters	11
Figure 9 Feature Importance	12
Figure 10 SHAP value	12
Figure 11 Snapshots of deployed streamlit app	14

Introduction

In today's urban landscape, the rental market plays a crucial role in shaping economic dynamics and housing accessibility. With rising rental prices and fluctuating market trends, both landlords and tenants face challenges in making informed decisions. This project aims to leverage machine learning to develop a predictive model for estimating monthly rental prices based on various property and location features.

Key highlights of the project include extensive exploratory data analysis (EDA) to understand the dataset, meticulous preprocessing to address data quality issues, and feature engineering to enhance predictive capabilities. Six supervised learning models were implemented, including Linear Regression, Random Forest, XGBoost, Support Vector Regression (SVR), LightGBM, and CatBoost. Additionally, four unsupervised learning models, namely KMeans, DBSCAN, PCA, and t-SNE, were used for clustering.

For deployment, the project employed Streamlit to create an interactive web application, allowing users to easily access and visualize the predictions. By harnessing advanced analytical techniques, this work seeks to provide actionable insights that can improve decision-making processes in the real estate sector.

Problem Definition

The objective of this project is to create a predictive model that estimates monthly rental prices based on various attributes of apartments. Understanding rental price dynamics is essential for both landlords and prospective tenants, as it aids landlords in setting competitive rental rates and assists tenants in making informed housing decisions.

Dataset Selection

For this analysis, the dataset selected is the Apartment Rent Data available on Kaggle. This dataset includes comprehensive information about rental listings, featuring attributes such as apartment type, location, size, and amenities, making it highly relevant for predictive modeling. With 99492 apartment entries each with 22 attributes, the dataset meets the criteria for complexity and size, providing sufficient data to apply advanced machine learning techniques. Furthermore, the dataset is well-documented and sourced from a reputable platform, ensuring its quality and reliability. Ethical considerations are also addressed, as the data does not contain sensitive personal information, allowing for responsible use in modeling rental price predictions.

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is a crucial step in understanding the dataset's characteristics, distributions, and relationships among features. The following analyses provide insights into the apartment rental data:

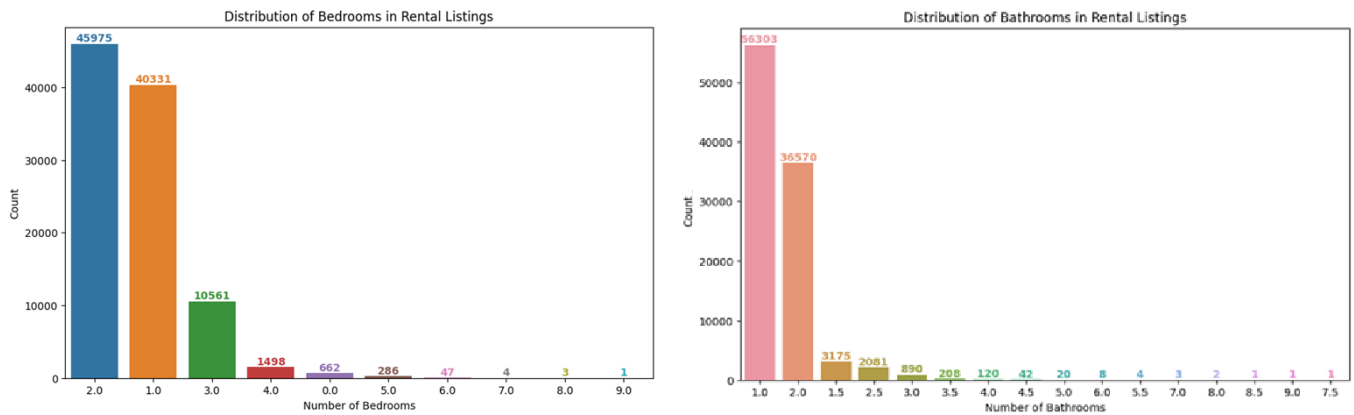


Figure 1 Distribution of Bedrooms and Bathrooms in Rental Listings

Both bathrooms and bedrooms features show similar distribution across listing with both have dominant values 1 and 2 followed by 3.

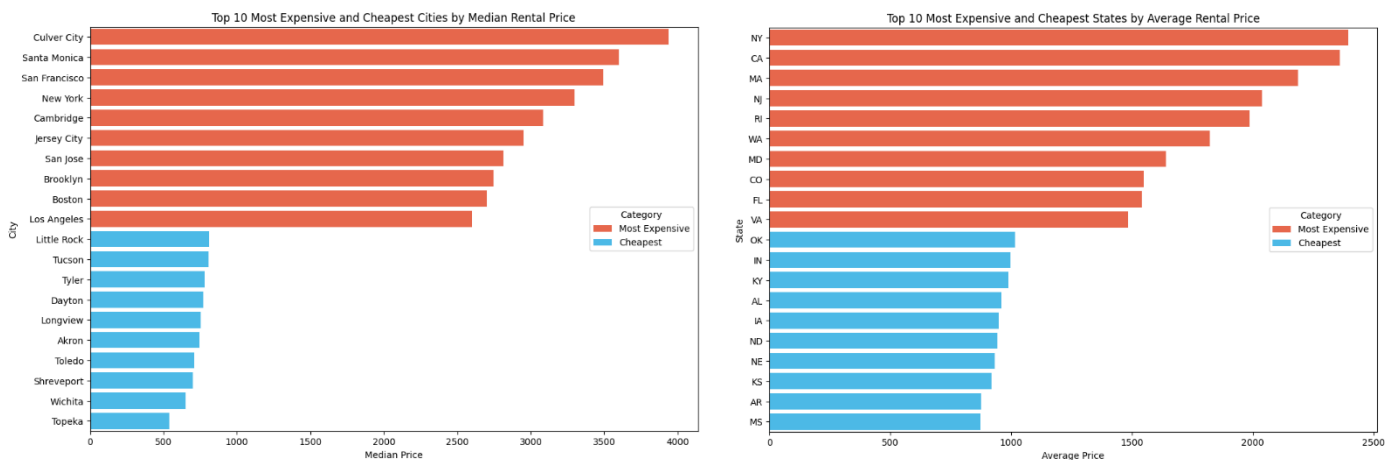


Figure 2 Top 10 Most Expensive and Cheapest States and Cities

The states of New York and California have similar average rental prices which is among the most expensive in the US. On the contrary, the states Arkansas and Mississippi are the cheapest.

In terms of median rental prices, Culver City is by far the most expensive city in the US to rent followed by Santa Monica, San Francisco, New York and then Cambridge. On the other hand, Topeka is the cheapest city in the US to rent followed by Wilchita and Shreveport.

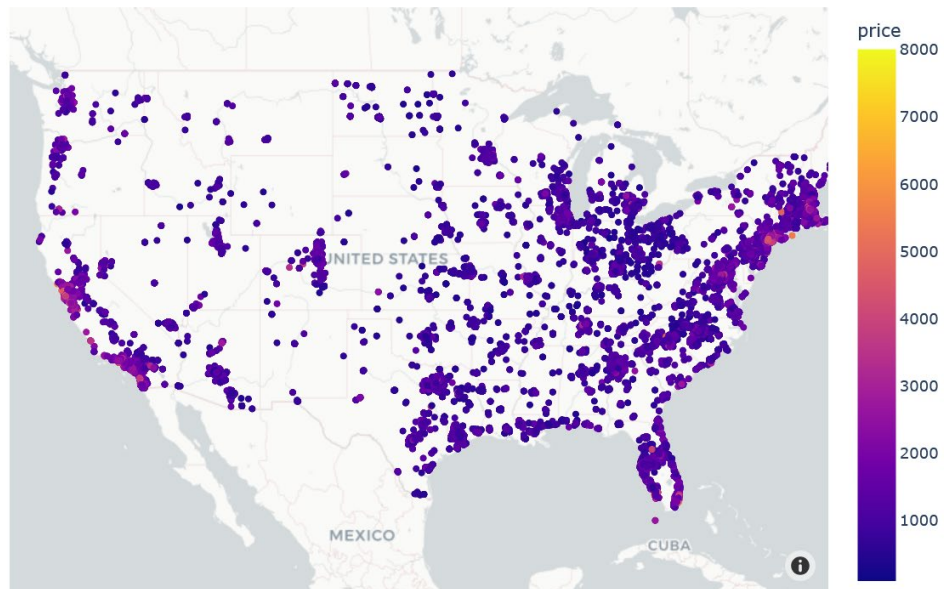


Figure 3 Geographical coordinates with rental price

It can be observed that prices tend to increase as we move towards the east or west of the country. From above, the indication is obvious that rental apartments near the ocean in the U.S. are higher priced compared to those further inland.

Data Cleaning and Selection

Data cleaning is a critical step in preparing a dataset for analysis and modeling. In this project, the rental dataset underwent a thorough examination to identify and rectify various quality issues.

Handling Missing Values

Most features with missing values had less than 1% missing except for amenities (16.12%), pets_allowed (60.73%) and address (92%). Based on this, the feature address should be removed and the rest of the missing values features can be handled.

For the categorical columns amenities and pets_allowed, the missing values were handled with placeholder imputation where the missing values were replaced with the placeholder text “missing”. For the numerical columns bedrooms and bathrooms, their missing values were replaced with their median value. Finally, any remaining rows with missing values were removed from the dataset.

Feature Removal

- **id:** It is just indexing column. No real impact on either insights or prediction.:
- **title and body:** These features have high number of unique value upto 95% of all rows which will not help in predicting prices.
- **price_display:** There is already a price column which is the cleaned feature of this column.
- **currency:** This feature has only one unique value which will not help in prediction.
- **address:** This feature is not necessary for this dataset as we already have the geographical coordinates of the listing.
- **source:** This feature was removed as for prediction purpose the source will not be necessary.

Further Cleaning

Before applying data cleaning, the dataset had 99189 rows and 15 columns.

Only entries with bedrooms less than or equals to 6 were kept. Also entries with bathrooms greater than 5 were removed. This was done so because there were no significant value counts for bedrooms more than 6 or bathrooms more than 5 which could potentially affect prediction capability. Then, both these columns were changed to integer types as there were float number present.

The feature fee is binary so its values were map from “0” to “No” and “1” to “Yes”.

The categorical features cityname and state have many unique values with less value counts. Removing those values would have resulted in significant decrease in the dataset size so the method of replacing those values with value counts less than a threshold value of 100 with placeholder text ‘Other’ was opted to minimize decrease in dataset size. On the contrary, unique values with value counts less than the same threshold value of 100 in the categorical features: category, has_photo, price_type and pets_allowed were removed completely.

From above data cleaning steps, the features category and price_type had been impacted. There were only a single unique value in both the features. So these two feature were also removed.

The features amenities and pets_allowed are multi-label or multi-valued feature. So these features were split into each separate columns with prefixes “amenities_” and “pets_allowed_”

Also the distribution of price column was highly skewed so only rows with price value less than or equals to 8000 were kept.

The resulting cleaned dataset had 98983 rows and 42 columns.

Model Development and Evaluation

This section outlines the development and evaluation of the machine learning models employed to predict monthly rental prices. A diverse set of algorithms were implemented to capture the complexity of the dataset effectively which was trained on GPU.

Feature Engineering and Preprocessing Pipeline

Before model training, a comprehensive preprocessing pipeline was established to ensure data quality and enhance model performance. The first step in the preprocessing pipeline was feature engineering where 3 new features were created. The “squarefeet_per_room” feature was derived by dividing the total square footage of the property by the combined number of bedrooms and bathrooms, providing a metric for space allocation per room. Additionally, two binary features, “is_northern” and “is_western”, were introduced to indicate the geographical location of the property based on latitude and longitude, respectively.

Categorical variables were encoded using one-hot encoding to facilitate their integration into the models. Additionally, feature scaling was performed on numerical features to standardize the data, which is particularly important for algorithms sensitive to feature magnitudes, such as Support Vector Regression.

Model Selection

A variety of machine learning models, including both linear and tree-based approaches. 6 models were implemented for supervised learning including Linear Regression, Random Forest, XGBoost, Support Vector Regression (SVR), LightGBM and CatBoost. Similarly for unsupervised learning 4 models: KMeans, DBSCAN, PCA and t-SNE were implemented for clustering. Each model was chosen for its ability to handle different types of data patterns and relationships, allowing for a comprehensive comparison of their predictive capabilities.

Data Splitting

To ensure robust evaluation of the models, the dataset was divided into three subsets: training, validation, and test sets. The training set comprised roughly 85% of the data, used for fitting the models. The validation set, accounting for roughly 6%, facilitated hyperparameter tuning and model selection. Finally, the remaining approximately 9% served as the test set to assess model performance on unseen data.

Evaluation Metrics

To measure the effectiveness of the supervised learning models, R-squared (R^2) was utilized as it indicates the proportion of variance explained by the model, offering a clear picture of its explanatory power. For the unsupervised learning, elbow and silhouette score approaches were utilized.

Cross-Validation

For supervised learning, K-fold cross-validation was implemented, setting k to 5, to enhance the reliability of evaluation. This method involved dividing the training data into five subsets, allowing each subset to serve as a validation set while the others were used for training. This approach mitigates the risk of overfitting and provides a more generalized performance estimate.

Performance Comparison

Before training, the dataset were processed using the preprocessing pipeline discussed above. For supervised learning the target feature was price and every other features were independent features which was processed using the preprocessing pipeline. For unsupervised learning the entire dataset was transformed using the preprocessing pipeline as there is no thing as target feature while clustering. After training and validating the models, their performances were compared based on the evaluation metrics.

Supervised Learning

Model	Fold 1 - R2	Fold 2 - R2	Fold 3 - R2	Fold 4 - R2	Fold 5 - R2	Average R2	Best R2
RandomForest	0.6020	0.6031	0.6000	0.6021	0.6040	0.6023	0.6040
SVR	0.1492	0.1489	0.1483	0.1504	0.1504	0.1495	0.1504
LinearRegression	0.6458	0.6467	0.6455	0.6463	0.6455	0.6460	0.6463
XGBoost	0.8080	0.8051	0.8054	0.8060	0.8007	0.8051	0.8080
LightGBM	0.7813	0.7892	0.7895	0.7829	0.7851	0.7856	0.7895
CatBoost	0.8101	0.8096	0.8088	0.8086	0.8092	0.8093	0.8101

Figure 4 Supervised Learning 6 models performance comparison

The evaluation of six supervised learning models on the rental price prediction task demonstrated varied performance across folds and metrics. The Random Forest model achieved an average R^2 of 0.6023, indicating a moderate ability to explain the variance in rental prices. In contrast, Support Vector Regression (SVR) performed poorly, with an average R^2 of only 0.1495, suggesting limited predictive power.

Linear Regression achieved a slightly better average R^2 of 0.6460, but the standout performers were the ensemble models. XGBoost led with an impressive average R^2 of 0.8051, while CatBoost closely followed with 0.8093, showcasing their effectiveness in capturing complex relationships within the data. LightGBM also performed well, with an average R^2 of 0.7856.

The best R^2 score model was Catboost in fold 1. So this model was used in deployment in later part of this report.

Unsupervised Learning

Before applying KMeans clustering elbow method and silhouette method were performed to find the best number of clusters. The result indicated 2 were the best number of clusters.

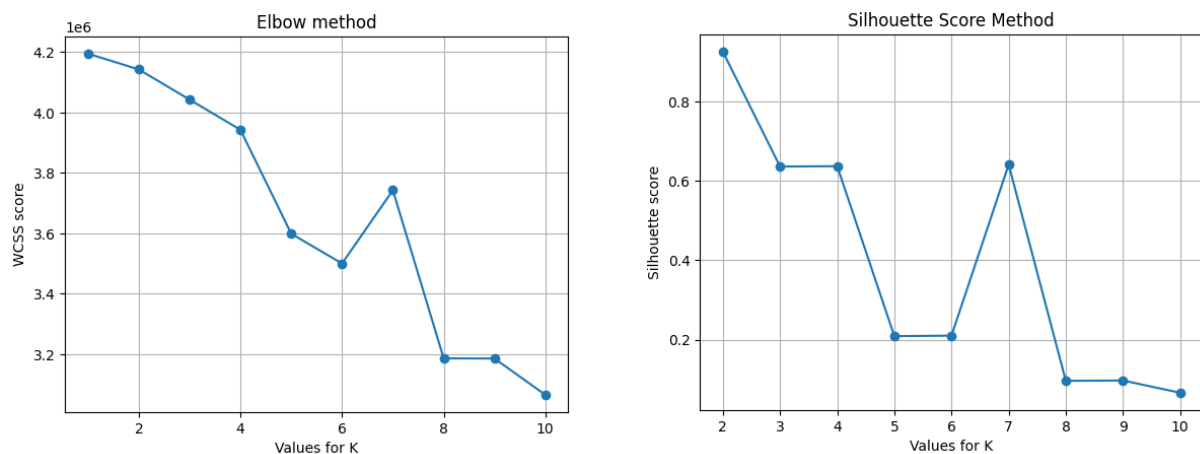


Figure 5 Elbow method and silhouette method with KMeans

Upon plotting the clusters, it was observed that most if not all entries were separated as a cluster. So there were no balance between clusters using KMeans.

DBSCAN algorithm with epsilon of 3 and minimum samples 6 resulted in more clusters than KMeans:

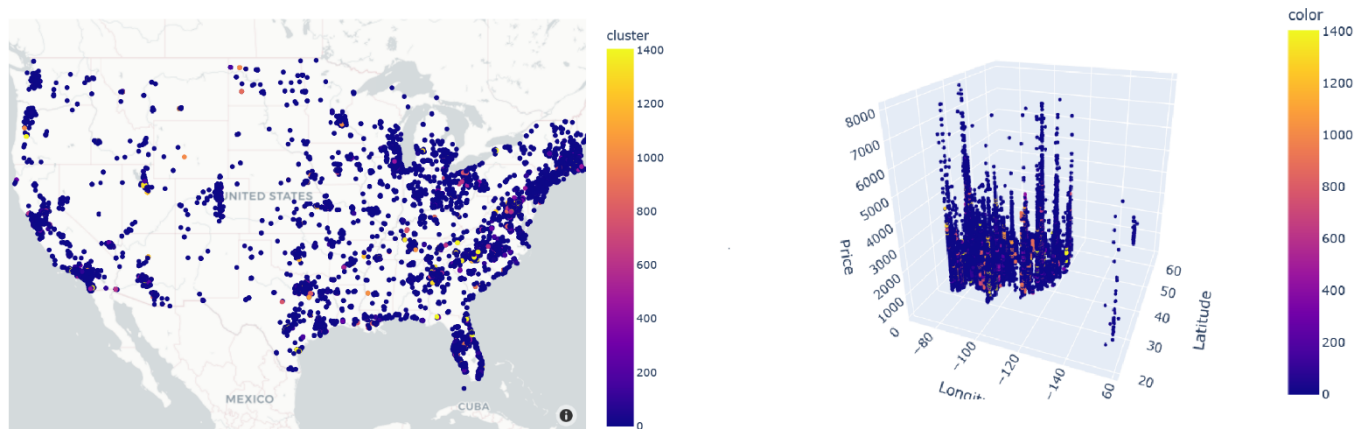


Figure 6 DBSCAN clusters visualization of rental listings

Then PCA was applied to decompose the features into 2 principal components. The resulting components had variance ratio of 0.12 and 0.05 which in total explains variance of just 0.17. Still the 2 principal components were used to visualize the DBSCAN clusters.

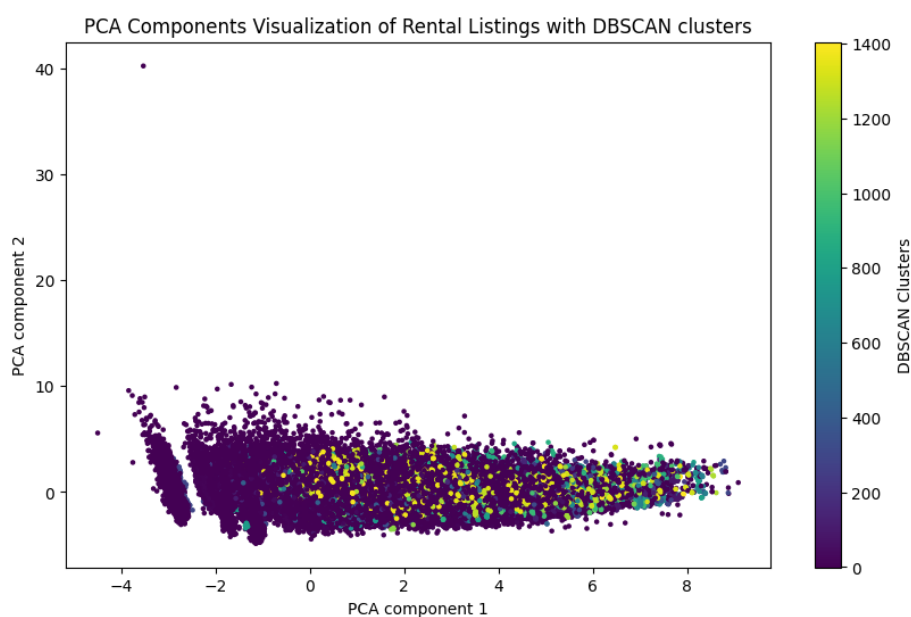


Figure 7 PCA Components Visualization of Rental Listings with DBSCAN clusters

Finally, t-SNE algorithm with number of components 2 was used to along with the same DBSCAN clusters:

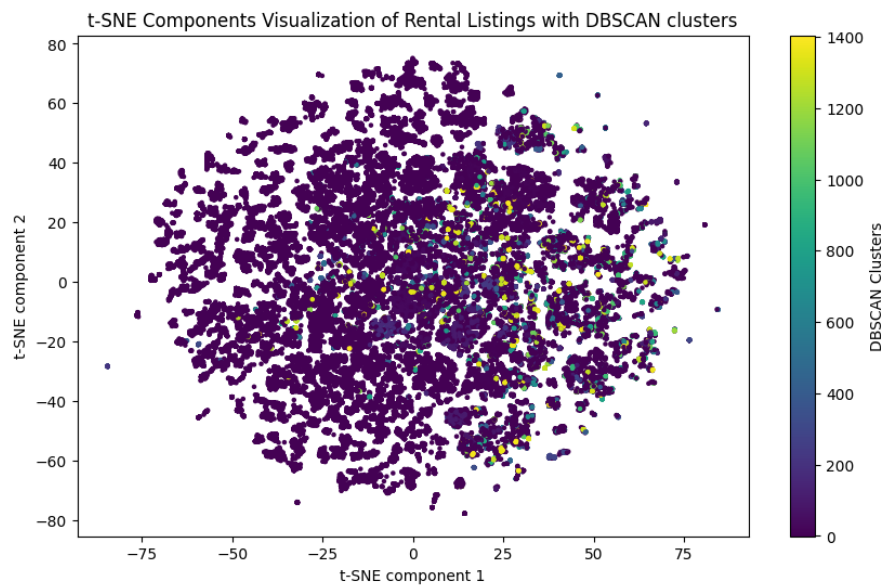


Figure 8 t-SNE Components Visualization of Rental Listings with DBSCAN clusters

Model Interpretation and Explainability

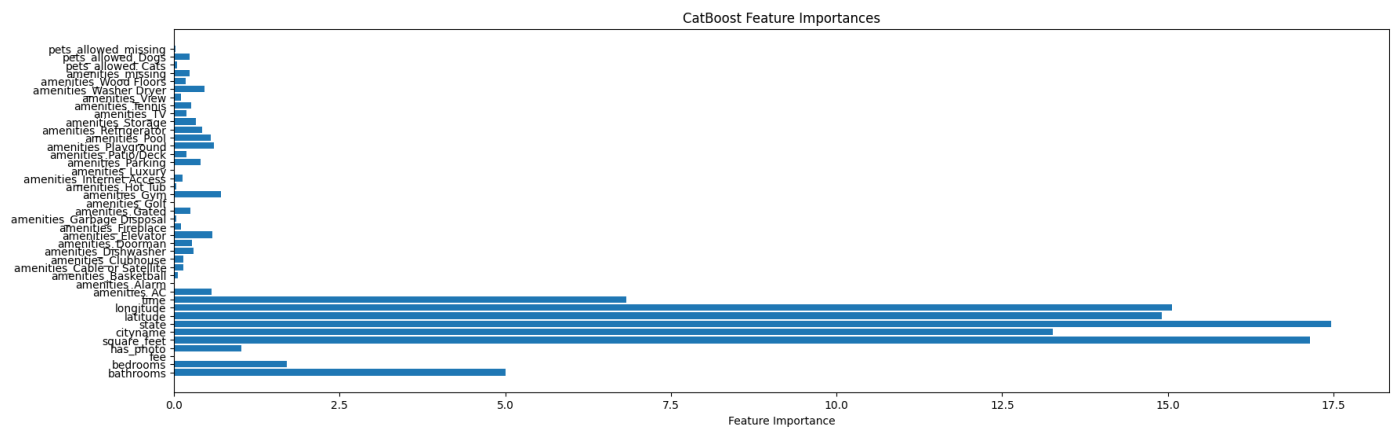


Figure 9 Feature Importance

Based on the feature importance, the state which you live in impacts the rent prices the most, followed by the area of the apartment. Then similarly, the important features comes out to be latitude and longitude. Remaining features which have high importance are time and then bathroom (more important than bedrooms). This insights suggest that the most important thing when predicting rent processes are the location features of the apartments.

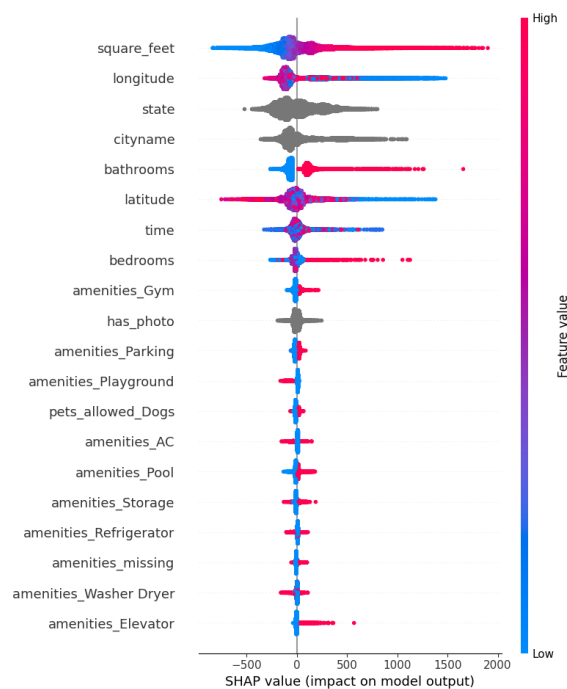


Figure 10 SHAP value

SHAP values also provide a similar importance of feature where location features like longitude, state and cityname are highly important.

Most important feature according to SHAP value is the area of the apartment. Higher square footage (red) is strongly associated with a higher predicted price (positive SHAP values). Conversely, smaller square footage (blue) is linked to a lower predicted price. The relationship is quite clear and strong.

The impact of longitude seems to be more complex and possibly non-linear. Some high longitude values increase the price, while others decrease it.

More bathrooms (higher values, red) tend to increase the predicted price.

Furthermore, most amenities, like having a gym, pool, or parking, have a positive impact on the price, though the effect is relatively small compared to the major factors like square footage.

Deployment Strategy

The deployment of the rental prediction model is a crucial step in making the insights gained from the machine learning process accessible to end users. The best performing model CatBoost was saved as a file and download for deployment pupose. Streamlit was used to generate the required application in a fast and easy manner. The complete project and steps to reproduce the result is outlined in the project github repo: <https://github.com/abhash-rai/US-Monthly-Apartment-Price-Prediction>

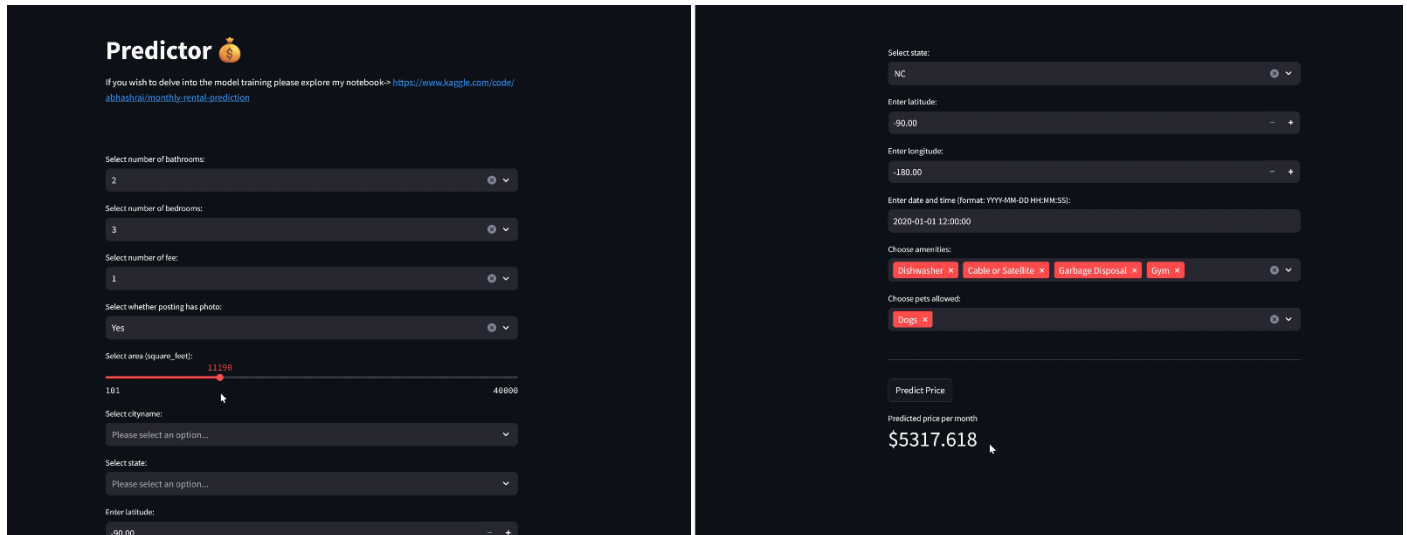


Figure 11 Snapshots of deployed streamlit app

Further Improvements

The application can be further improvedby implementing a system that tracks key performance indicators (KPIs) such as prediction accuracy and response times. After predicting the rent price an additional input field can be presented to include the actual price of the apartment which would be stored in a database or a csv file for future reference or further training of the model.

Handling Concept Drift

Concept drift refers to the changes in the underlying data patterns over time, which can affect model accuracy. To mitigate this, a strategy is in place for handling such shifts. This includes:

- **Periodic Model Evaluation:** The model's performance will be regularly assessed against new data to identify any significant drop in accuracy.
- **Retraining Procedures:** If a decline is detected, the model will be retrained using the latest dataset to adapt to new trends in rental pricing.
- **Version Control:** Implementing version control for the models will help in tracking changes and ensuring that the best-performing model is in use.

Conclusion

In summary, this project successfully developed a predictive model for estimating monthly rental prices, demonstrating the practical application of advanced machine learning techniques. Through thorough exploratory data analysis and thoughtful feature engineering, key factors influencing rental prices were identified. The model's performance, evaluated through various metrics, indicates its effectiveness and reliability. Additionally, the insights gained underscore the importance of these predictions in informing decision-making for both landlords and tenants. Future work could enhance model accuracy by incorporating more diverse data sources and advanced algorithms, further improving its applicability in real-world scenarios.