Predicting Monthly Rent Prices of Provo Apartments

Jackson Baxter, Will Clayton, Cooper Johnston
CS270, Fall 2024

Abstract

Students often struggle to find affordable housing that meets their needs within Provo due to limited budgets and high demand near universities such as BYU and UVU. To help students make informed decisions, we used machine learning to predict apartment monthly rent prices based on features like amenities, size, and location. Our dataset included 66 data points collected manually from apartment websites, and we tested the data on five models: Lasso Regression, Decision Tree, Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), and Bayesian Ridge. Of these, KNN performed the best after we normalized the data, achieving an R^2 of 0.82. Principal Component Analysis (PCA) also improved our model efficiency by reducing the dataset complexity. This project demonstrates that price can be predicted based on what an apartment offers, and this prediction is useful to compare to the real price in order to estimate how good of a value each apartment is.

1. Introduction

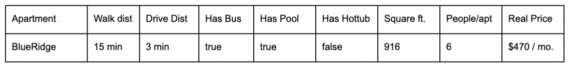
The city of Provo is heavily populated with students due to the close proximity to BYU and UVU. As such, finding reasonably-priced housing in Provo can be difficult for many students, especially considering the low budgets that students have available to them as they attend school full time. According to rentcafe.com, roughly 61% of housing in Provo is renter-occupied as opposed to owner-occupied, meaning that the majority of Provo residents are living in apartments.

Although many apartment complexes exist near BYU campus, students may struggle to know which apartments are a good price for what they have to offer. Monthly rent prices vary greatly depending on many factors, from as low as ~$300/mo. to well over ~$1500/mo. Despite having options, students often express dissatisfaction with their living situations. In order to help students make well-informed decisions about where they choose to live, we wanted to train machine learning models to predict the monthly rent based on the amenities and other desirable qualities that apartments provide. Then, that prediction can be compared to the actual price that the apartment charges for rent. If the prediction is below what the apartment actually charges, it suggests that it may be overpriced. On the other hand, if it's above what the apartment actually charges, it may be underpriced. Getting a feeling of what apartments are overpriced or underpriced can help students to find good deals, or get out of current suboptimal living situations.

2. Methods

2.1 Initial Features

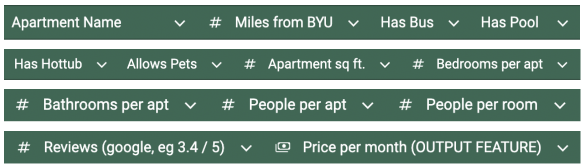
In the original project proposal, several features were listed that we ended up collecting data for and using in our final dataset. The originally-proposed features are shown in the table below:



Apartment	Walk dist	Drive Dist	Has Bus	Has Pool	Has Hottub	Square ft.	People/apt	Real Price
BlueRidge	15 min	3 min	true	true	false	916	6	$470 / mo.

We kept most of the proposed features since they seem like they would all be helpful metrics in determining the amount that an apartment charges for monthly rent. Amenities such as pools, hot tubs, and a shuttle bus that goes to campus would all make an apartment complex more desirable and therefore able to charge more in rent. Similarly, square footage and people per apartment play a large role in determining how much living space a resident would have, and would likewise increase the rent price.

We did, however, make several modifications and additions to the features before we began gathering data. We combined "walk dist" and "drive dist" into a single feature and decided to measure it as a certain number of miles from BYU instead of a time to reach BYU. Also, in addition to people per apartment, we decided to include bedrooms per apartment, bathrooms per apartment, and people per room (simply a combination of people/apartment and bedrooms/apartment, but potentially a useful input feature nonetheless). Finally, we added the google reviews (out of 5 stars) of each apartment complex. While this information was not listed on individual apartment websites, it was easily accessible via Google search, and probably an accurate and helpful feature since hundreds of people have left reviews for most of the apartment complexes we included. The full list of final features is shown below, separated into multiple lines for ease of reading:


Apartment Name | # Miles from BYU | Has Bus | Has Pool

Has Hottub | Allows Pets | # Apartment sq ft. | # Bedrooms per apt

Bathrooms per apt | # People per apt | # People per room

Reviews (google, eg 3.4 / 5) | Price per month (OUTPUT FEATURE)

2.2 Data Gathering

In order to gather enough data to properly train the desired models and accurately predict housing prices, we visited several websites directly and manually collected data by inputting information into a spreadsheet. We figured that we would need at least 60-100 data points in order to properly provide the machine learning algorithms with enough data to discover meaningful patterns. Thankfully, most modern apartment complexes have their information easily accessible on the internet, usually on their own website. So, we used Google Search to discover a wide variety of apartment complexes, as well as online lists of Provo apartments such as BYU's list of approved off-campus housing (https://och.byu.edu/byu-fall-2024-contracted-housing). With specific apartment names in mind, we could simply type the name into google and find the website corresponding to the apartment complex. The websites almost always included a "floor plans" section with all of the information we needed to collect. Also, we were often able to find multiple floor plans for the same apartment complex, which were useful data points because they had varying prices, square footage and number or people/rooms, despite having the same amenities.

In the end, we collected 66 data points. We would have done more if time permitted, but it started getting very time consuming trying to find more apartment data, and we needed to move on and begin the next phase of the project. We believe that we collected enough data for some solid results and patterns to emerge as we use the data.

2.3 Dataset Augmentation

We needed to augment the raw data that we collected to make it usable for our models. There were several missing values (for example, if we couldn't find square footage for a given apartment, or the google rating) and some columns that weren't yet formatted correctly for our purposes. In order to clean the data, we started by dropping the "Apartment Name" column, which is a non-predictive column, meaning we only had that column for our own clarification and it (most likely) doesn't play into the price that the apartments charge. Next, we converted the categorical values for columns "Has Bus", "Has Pool", "Has Hottub", "Allows Pets" to be 1/0 instead of TRUE/FALSE. Next, for the output column (price/month) we removed extraneous $ symbols and commas, and made sure to cast it to a float. This way we ensured that the target is numeric for regression. We then separated into X and y for our testing by dropping the output column for X and including only the output column for y. Finally, we handled the missing values using the fillna function and passing in X.mean, simply inserting the mean from the rest of the values in the column as an estimation as to what the value should be for the row where it's missing.

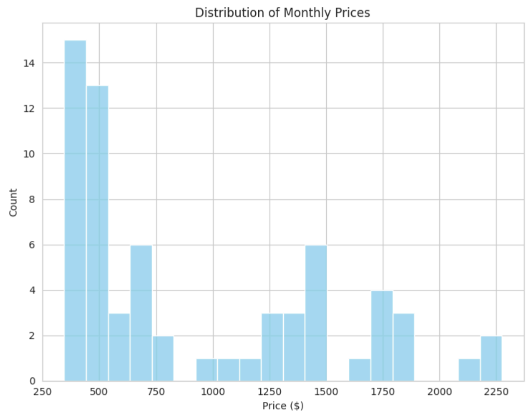
2.4 Feature Correlations

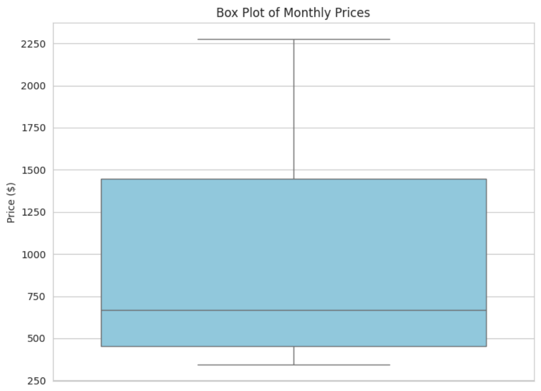
Various graphs and plots are displayed on the next page, summarizing the trends in monthly rent prices for Provo apartments, and highlighting how certain features affect pricing. The distribution of monthly prices (top-left) shows a right-skewed distribution, with many apartments priced between $400 and $600, but some priced much higher. The box plot (top-right) indicates a median price of around $700 and a broad range of prices, with some exceeding $2000.

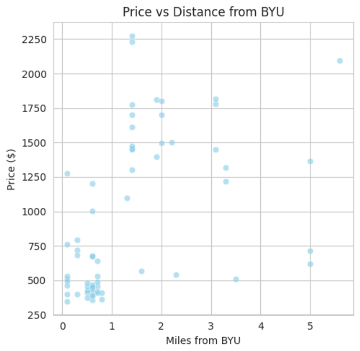
Key factors that influence rent include the number of bedrooms and bathrooms, as seen in the middle row. Apartments with more bedrooms are generally cheaper per resident, and single-bedroom units tend to have much higher prices.

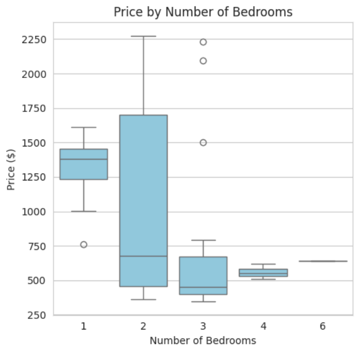
Amenities such as a shuttle bus, pool, hot tub, and pet allowance also seem to relate to price. While allowing pets appears to have a strong positive correlation with rent prices, the correlation for other amenities is not nearly that strong. For example, having a pool is actually correlated with cheaper rent prices even though a pool is a desirable amenity. This is probably because the apartments that have a pool tend to be large ones with many residents. Since they have many residents, they have more residents per apartment, which as mentioned, means cheaper rent per resident.

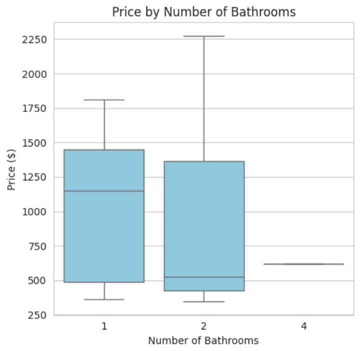
Many other insights may be discovered from the data visuals below:

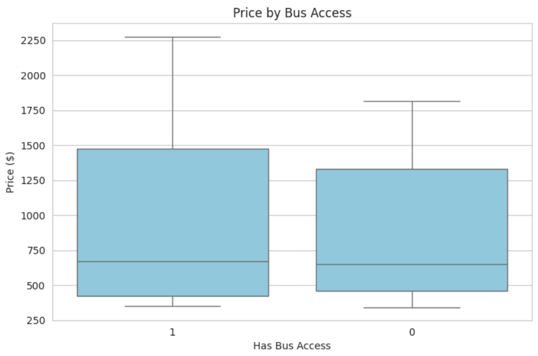

Distribution of Monthly Prices
Count
0
2
4
6
8
10
12
14
250
500
750
1000
1250
1500
1750
2000
2250
Price ($)

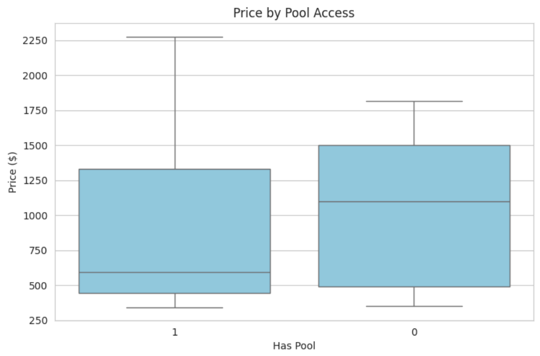

Box Plot of Monthly Prices
Price ($)
250
500
750
1000
1250
1500
1750
2000
2250

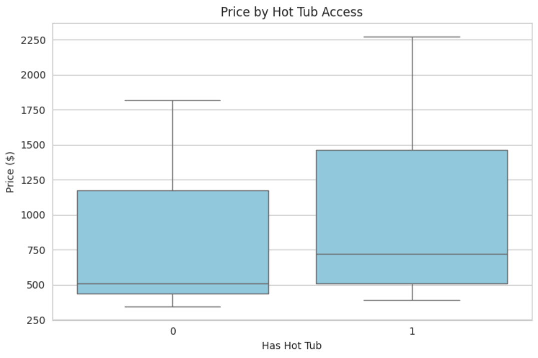

Price vs Distance from BYU
Price ($)
250
500
750
1000
1250
1500
1750
2000
2250
0
1
2
3
4
5
Miles from BYU

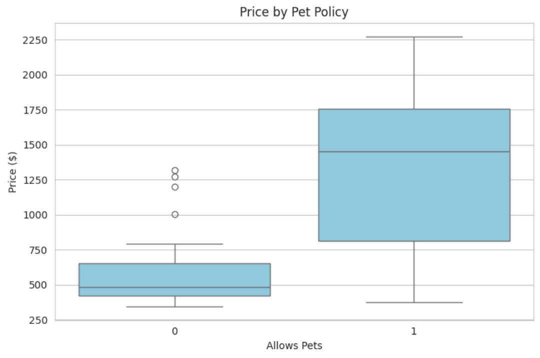

Price by Number of Bedrooms
Price ($)
250
500
750
1000
1250
1500
1750
2000
2250
1
2
3
4
6
Number of Bedrooms


Price by Number of Bathrooms
Price ($)
250
500
750
1000
1250
1500
1750
2000
2250
1
2
4
Number of Bathrooms


Price by Bus Access
Price ($)
250
500
750
1000
1250
1500
1750
2000
2250
1
0
Has Bus Access


Price by Pool Access
Price ($)
250
500
750
1000
1250
1500
1750
2000
2250
1
0
Has Pool


Price by Hot Tub Access
Price ($)
250
500
750
1000
1250
1500
1750
2000
2250
0
1
Has Hot Tub


Price by Pet Policy
Price ($)
250
500
750
1000
1250
1500
1750
2000
2250
0
1
Allows Pets

3. Models

To evaluate the performance of various machine learning algorithms, we trained and tested five different models: Lasso, Decision Tree, Multilayer Perceptron (MLP), K-Nearest Neighbors (KNN), and Bayesian Ridge. Each model gave unique perspectives of the data, and we measured their performance using metrics such as Mean Squared Error (MSE) and R^2, which we report later in the results section. Below are the explanations for the implementation behind each model.

3.1 Lasso

Lasso regression is a linear regression that applies L1 regularization to penalize large coefficients, which reduces the complexity of the model and helps prevent overfitting. It also performs feature selection by lowering the coefficients of less important features to zero. We trained the model on our dataset and evaluated it using MSE and R^2. This method was particularly useful to identify whether all features in the dataset were meaningful or if some could be ignored in future iterations.

3.2 Decision Tree

The Decision Tree model uses certain decision rules to predict rent prices by splitting the features. Unlike linear models, Decision Trees can identify non-linear relationships in the data, making it a good candidate since there's a wide variety of amenities and numeric attributes in our dataset. After training, the Decision Tree model was evaluated on both training and testing data to ensure that it generalizes well and does not overfit. This method is especially useful in capturing patterns like how specific amenities might impact rent prices.

3.3 MLP

The Multilayer Perceptron was trained to model the more complex, non-linear relationships within the dataset. It uses hidden layers to identify deeper patterns that simpler models might miss. Although neural networks often require large datasets to perform well, we still wanted to see how well it could adapt to our smaller dataset. We used standard metrics such as MSE and R^2 to find its accuracy.

3.4 KNN

KNN predicts rent prices based on the average rent of the k most similar data points in the dataset, meaning it would work well for discovering clusters. We trained the model with default parameters and evaluated its performance.

3.5 Bayesian Ridge

Bayesian Ridge Regression adds Bayesian inference to the linear regression, offering uncertainty estimates for its predictions. Like Lasso, it regularizes the model to prevent overfitting but does it probabilistically. Using this method helped us to see how well the features contribute to the rent prices. Bayesian Ridge was evaluated for both training and testing sets to ensure consistency.

4. Results

4.1 Initial Results

The initial results showed that some models worked better than others. The Lasso Regression performed the best, with a Train R^2 of 0.8528 and a Test R^2 of 0.7761. This means it explained most of the variance in rent prices and worked well on both the training and test data. The MSE values for training (44,424) and testing (84,254) were also fairly low, which shows that it made good predictions.

The Decision Tree model gave decent results with an RMSE of 299.66, but decision trees can sometimes overfit, so we'd need to check more to see if it was too focused on the training data.

MLP did not seem to work as well, with a Train R^2 of -0.1585 and a Test R^2 of 0.0773. It also showed a warning that it didn't finish optimizing, which might explain the poor results. Similarly, KNN and Bayesian Ridge struggled. KNN had an MSE of 400,869, while Bayesian Ridge had basically no predictive power, with a Train R^2 of 0.0 and a Test R^2 of -0.0196.

Overall, Lasso gave the best results for the initial runs, while the others didn't do as well, possibly because of the small dataset or the complexity of the models.

4.2 Improvement Efforts

To improve the performance of our models, we applied feature scaling and dimensionality reduction. Using StandardScaler, we scaled the features to ensure that all input variables had similar ranges, which is especially important for models like KNN and neural networks. We also used Principal

Component Analysis (PCA) to reduce the dataset to its 10 most important components. We did this to help simplify the dataset while keeping most of the useful information, potentially reducing noise and increasing accuracy. For individual models, we did some more specific optimizations:

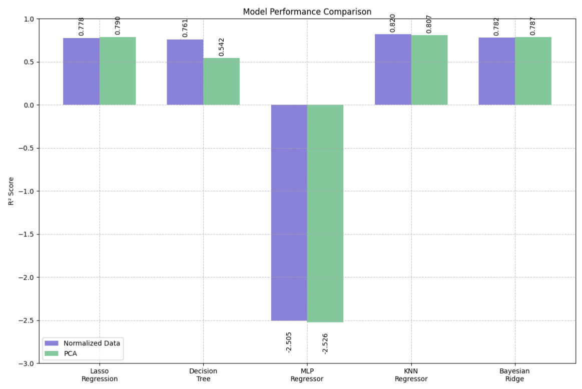
1. **Delta Rule Regressor**: A custom implementation using a different loss method provided much better results, achieving a Test R^2 of 0.7834.
2. **Decision Tree**: We did a grid search to find the best hyperparameters, such as maximum depth, splitting criteria, and minimum samples per leaf. The optimized Decision Tree was easier to interpret, though its Test R^2 of 0.6506 shows that there might've been some overfitting.
3. **MLP**: We tested hyperparameters like learning rate and momentum through grid search.
4. **KNN**: We again tested hyperparameters for the number of neighbors, weight strategy, and distance metric, which definitely improved the performance.
5. **Bayesian Ridge**: We did grid search for hyperparameters like alpha and lambda, resulting in a Test R^2 of 0.8486, which is a bit better than the default configuration.

4.3 Final Results

After testing all the models, here are the final results we achieved:

1. **Lasso Regression**:
    - Using normalized data: Test R^2 = 0.7778
    - Using PCA: Test R^2 = 0.7899
2. **Decision Tree**:
    - Using normalized data: Test R^2 = 0.7614
    - Using PCA: Test R^2 = 0.5422
3. **MLP Regressor**:
    - Using normalized data: Test R^2 = -2.5052
    - Using PCA: Test R^2 = -2.5261 (Convergence issues likely caused poor performance. It would probably work better if we had more data.)
4. **KNN Regressor**:
    - Using normalized data: Test R^2 = 0.8195
    - Using PCA: Test R^2 = 0.8073
5. **Bayesian Ridge**:
    - Using normalized data: Test R^2 = 0.7821
    - Using PCA: Test R^2 = 0.7874
6. **Comparison of Averages**
    - Using normalized data: Average Test R^2 = **0.1271**
    - Using PCA: Average Test R^2 = **0.0801**

It appears that KNN performed the best overall, achieving the highest Test R^2 (0.8195) with normalized data. It increased greatly in accuracy when we did the normalization of the data. Lasso regression with PCA was the next best, and gave a good balance between being simple and also accurate (Test R^2 = 0.7899). In general, PCA helped simplify the dataset without a major loss in accuracy, but its benefit varied by model. The average for each model was very low because of the poor neural net score. An ensemble using these machines would be very ineffective, especially if using an even vote like bagging.



5. Conclusion and Future Work

Our study demonstrated the effectiveness of machine learning models in predicting apartment rents based on amenities and other qualities. Normalizing the data consistently improved model performance, with KNN achieving the highest accuracy (Test R^2 = 0.8195). Dimensionality reduction through PCA simplified the dataset without a significant loss in accuracy for most models, highlighting its potential for reducing noise and improving interpretability.

However, some limitations emerged, including the MLP Regressor's convergence issues and potential overfitting in the Decision Tree model. These challenges suggest that increasing the dataset size and diversity could help improve model generalization and stability.

To build on this work, we propose expanding the study in two key directions:

1. **Broadening the Geographic Scope:** Extending the analysis to include other cities in Utah, such as Salt Lake City, Draper, and Ogden, would provide a broader context for rent predictions. These areas have diverse populations and housing markets, which could make the models more robust and applicable across the state.
2. **Incorporating More Housing Types in Provo:** While our current study focuses on apartments, including other housing types such as townhomes, single-family rentals, and student housing complexes could provide a more comprehensive understanding of the local market. This would help students and other renters compare a wider range of living options.
3. **Looking at Different Property Management Companies and Other Features:** In addition to previously mentioned features, it is worth noting that a majority of properties in Provo are owned by a relatively small number of property management companies in Provo, including Nxt Property Management and the infamous Redstone Property Management. It is possible that certain companies tend to have higher pricing as well.

By expanding the geographic scope and housing types, we can refine our models to better serve renters and stakeholders across Utah, offering more actionable insights into housing affordability and value.

References

RENTCafé. (n.d.). *Provo, UT rental market trends*. Retrieved December 4, 2024, from https://www.rentcafe.com/average-rent-market-trends/us/ut/provo/#:~:text=Provo%2C%20UT%20Occupied%20Housing%20Units&text=occupied%20Households:%2039%25-,End%20of%20interactive%20chart.,39%25%20are%20owner%2Doccupied