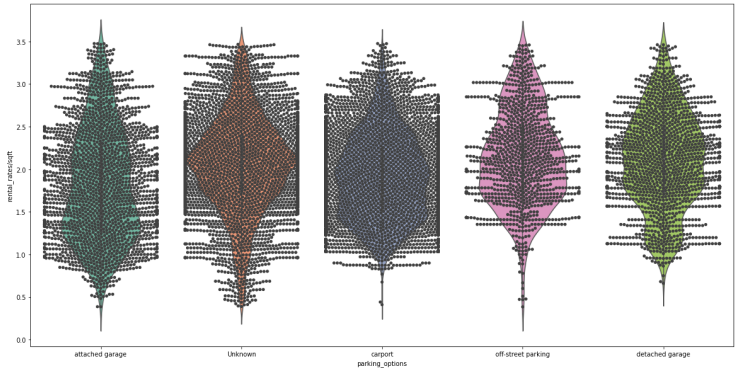
Predicting rental prices using key features obtainable from housing listing by easily scalable models

1. Exploratory Dataset Analysis

Rental house price is always a crucial economic factor and is affected by different factors like current supply and demand, current politics, and housing conditions. Fromhttps://www.kaggle.com/austinreese/usa-housing-listings/code, we choose "USA Housing Listings"as our dataset to explore potential factors that will influence rental housing prices. The dataset contains information about the 2020 monthly rental price of different types of houses across different cities in the United States with 33085 observations. The dataset was scrapped on Criagslist posting data on various Criagslist sites during 2020. Craigslist is an online platform where anyone can make a posting about any service or good that they would like to provide for a said price.

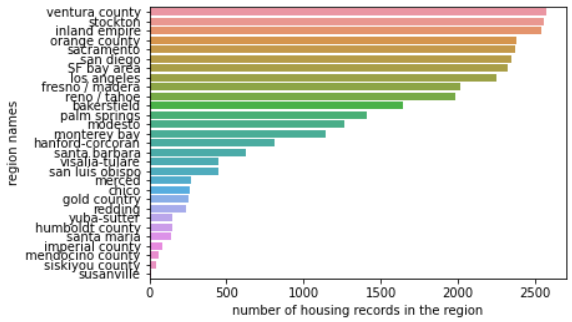
Among all the areas within the dataset, we decided to analyze particularly those from California because of not only the tremendous rise in housing prices, but also because of that we study at San Diego and wish to make some contributions in understanding our community through allowing more people to understand factors that can influence housing rental price.

The dataset contains 22 columns - id, url, region, region_url, price, type, sqft, beds, baths, cats_allowed, dogs_allowed, smoking_allowed, wheelchair access, electric_vehicle_charge, comes_furnished, laundry_options, parking_options, image_url, description, lat, long, and state. The variables being utilized in our assignment include "region", "type", "laundry_options", "parking_options", "description" as objects, "price", "sqfeet", "beds", "cats_allowed", "dogs_allowed", "smoking_allowed", "wheelchair_access","electric_vehicle_charge", "comes_furnished" as integer, "baths" as floats to explore different predictive models using different features.



Distribution of parking_options against price/sqfeet.

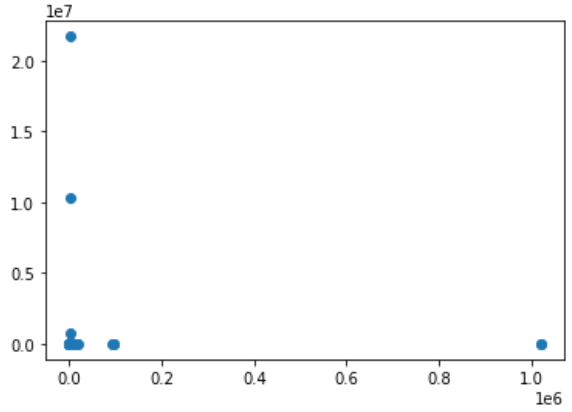
After exploring statistical analysis, we find that for "price", it has a mean of 8.83k, a median of 1036, a standard deviation of 4.46m, a minimum of 0, and a maximum of 2.77b. For "sqfeet", it has a mean of 1.06k, a median of 949, a standard deviation of 19.2k, a minimum of 0, and a maximum of 8.39m. For "baths", it has a mean of 1.48, a median of 1, a standard deviation of 0.64, a minimum of 0, and a maximum of 75. For "beds", it has a mean of 1.91, a standard deviation of 3.49, a median of 2, a minimum of 0, and a maximum of 1100.



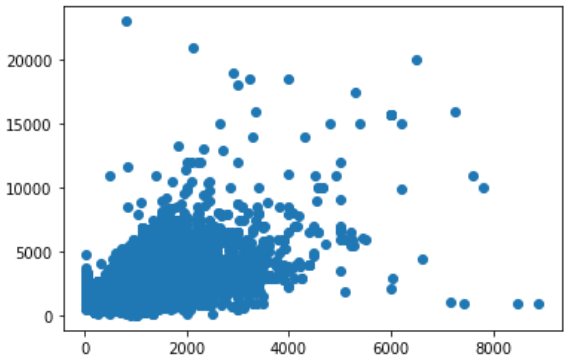
The above figure shows the number of housing records in each area in California. This is useful when we apply our Ridge model on each of the area in California.

For data cleaning, we find that there are 5067 null values in column “laundry_options” and 8503 null values in column “parking_options”. We fill null values in “laundry_options” with “no laundry on site” and null values in “parking_options” with “no parking”. To eliminate outliers, we tried to use interquartile range to define the upper and lower outlier limits, but the result still turned out to be quite sparsely distributed. Thus, we use a histogram plot to determine the upper boundary for the price to be $10000.

Before removing outliers in sqfeet and prices, the scatter distribution is like the plot below.



After deciding removing outliers in prices and sqfeet (area of rooms in square feet) so that no house is more than 25000 sq ft and 10000 dollars. The scatter distribution became what is shown below.



It’s interesting to find out that in California, the Visalia-Tulare region has the highest monthly rent ($24269.9) and imperial county has the lowest monthly rent ($879.3). Housing that allows pets usually has a higher rent than housing that doesn’t allow pets in every region.

2. Predictive task

Our predictive task is to predict housing prices in California based on different features listed in the dataset. We decide to predict housing prices because we believe that housing price is always the priority of renters' concerns, and thus we would like to perform a meaningful prediction that is useful in reality and can do real good to people.

Especially when considering since the pandemic, housing prices have been a hot topic, having an easy to assemble and accessible tool for future renters to evaluate a fair market price for their future residence is essential to level the playing field between renters and homeowners. As students studying in the State of California, housing cost is one of the biggest expenses in our budget, having a model that would allow students who usually have little to no knowledge about the current state of the housing market to have an easier time to call a place their home during their college studies.

The reason that our team chose to use the State of California as the main focus is that the state has a large population and a generally larger sample of housing stock that are available in the dataset. The team's prior knowledge with the State of California and the housing market is also a factor that was taken into consideration when the decision was made as it allows us to better evaluate the accuracy of the model itself.

We believe that the dataset would be a good fit for the predictive task because the dataset contains various features, for example sqft of the property, bedroom and bathroom count. Such features listed in the dataset are features that our team believe a renter would evaluate and value when they are renting a new home. Most of the features in the dataset are numeric or categorical values, meaning that these values would be easily utilized. There is also a description column, which contains a detailed description of the posting.

Although the dataset was a bit dated (2 years ago), we believe that such a model would still be extremely valuable because it was after the big interruption that was caused by the pandemic. The pandemic had set a new paradigm in housing where people valued more on working space as more people are working from home as a result of the COVID-19 pandemic. Such a preference could still be seen today, especially when compared to the pre-pandemic era.

To evaluate our results and compare different models, we would use different scoring methods. In our research, we use a variety of scoring methods, which would be further elaborated in the model section.

Alongside different methods of evaluating the model, our team chose to use the train test set split method because it is the most straightforward one. Our train-test split evaluation uses the sklearn.preprocessing module to split the data into sets. The train set contains the most data, at 80% is the dataset that would be fitted to the model. The test set, which contains less data compared to the train set, takes the remaining 20% of the data to allow us to evaluate the data.

3. Model

To achieve our goal of having a solution of the predictive task that could be easily implemented by the general public, our team decided to use the general model that is in the sci-kit learn package. The sci-kit learn package is commonly referred to and known as sklearn, the package could be imported by using Python, a commonly known language from coding and data related research.

Before any models would be able to be used, our team further processed the original dataset and to see which variable could be realistically given when a person is finding their next home. After careful consideration, we believe the following features could be realistically given by the users that are also featured in the dataset:

- Metropolitan region that the home is located
- Type of residence (e.g.: house, apartment etc.)
- Size in square feet
- Number of bedrooms
- Number of bathrooms
- Whether cats are allowed
- Whether dogs are allowed
- Whether smoking is allowed
- Whether the home has wheelchair access
- Whether the home has electric vehicle charging access
- Whether the home comes furnished

- Type of laundry the home comes with (e.g.: in units, in the same complex etc.)
- Type of parking available (e.g.: off-street, on street etc.)
- Description of the listing

To feed the features into the model, our team uses a couple of transforming techniques. To transform everything other than the description, we used sklearn's OneHotEncoding preprocessor which allows the dataset's categorical values to become numeric categorical values. To allow the incorporation of the description into the model, our team uses the Term Frequency Inverse Term Frequency to filter out useful words and count the words among the dataset. Such a process was done using the sklearn's TF-IDF vectorizer.

Among all the models that the sklearn package provides, we decided to narrow our focus in 3 main type of models – Linear Regression, Ridge Regression and Lasso Regression (known as sklearn.linear_model.LinearRegression, sklearn.linear_model.Ridge, sklearn.linear_model.Lasso in the sklearn package).

To score the results that the model produces, our team opts to use mean square error and the r2 score.

The mean square error score, calculated by the formula: $\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2$, is a metric that represents the square of the average error that the prediction of the model would make.

The r2 score, calculated by the formula: $1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y_i})}{\sum_{i=1}^{n}(y_i - \bar{y_i})}$, is a metric that represents how much percentage of correctness that the model predicts when compared to the mean of the correct output.

To establish the baseline for the further comparison of our model, we decided to create a basic linear regression model, such a model would just fit the feature that we had listed above.

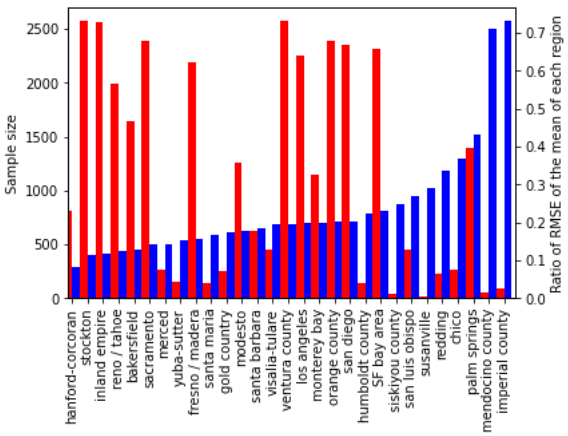
Ridge regression and lasso regression provides a significant improvement when compared to the baseline model. Both models allow for more degrees of control from the features as we are able to put a penalty term to the feature that is less relevant. The difference between the two models is that ridge regression would not completely eliminate the feature, while lasso regression would.

During our model development phase, we found that both models performed better than our baseline linear regression model. Our linear regression model achieves a mean square error of XX on the test set, compared to our ridge regression model of 158439 and lasso regression model of 335475. The ridge and lasso regression model also performs better when compared using the r2 score metrics, where the linear regression model has a r2 score of 0.4 on the test set, the lasso has a score of 0.81, and the ridge has the score of 0.64. From the above results, above the two better models that our team utilized, ridge performed better, this shows that the features that we picked are indeed essential to the predictive task.

However, even with the best model out of the three, our team is not satisfied with the results. The model still produces an average error of $400 when compared to the average mean rent of $1885 in the dataset is still unacceptable for general use. Therefore, our team decided to find the reason behind the model's shortcomings.

One of our hypotheses is that the region of the dataset would be a main reason for such a failure because regions are unique and if there is not enough data in each of the regions, the model cannot be trained on reliably enough to learn the pattern well enough.

To realize our theory, our group decided to build ridge regression models for every region that were listed and to see whether there is a correlation between a larger sample size and better model performance.



The above figure shows the model test set performance and the sample size of the results. Note that the score is inversed r^2 score, meaning that the lower the score the better.

The results shows that the model is easily affected by the sample size of each region, if we have a bigger sample size in each region, we are likely to achieve to better accuracy in the model.

4. Literature

In “New Insights into Rental Housing Markets across the United States: Web Scraping and Analyzing Craigslist Rental Listings”, the researcher scraped data from eleven million Cragistlt rental housing. The researchers stress the current problem with real estate and state “rentals compose a significant portion of the US housing market, but much of this market activity has been little understood because of its informal characteristics and historically minimal data trail” (Boeing & Waddell, p.g 468) and assess affordability by calculating rent burdens and proportion of listings. They find that only 37% of regions are below the corresponding fair market rent. “However, researchers find out that some metros like New York and Boston are only in the single-digit percentages, suggesting significant affordability challenges to local practitioners”(Boeing & Waddell, p.g 468-469).

Our dataset is an existing dataset from Kaggle. The dataset aims to perform experimental analysis on the United States as a whole and gather data by scraping all rental housing information during 2020. The dataset was used to build a linear regression model and predict rental housing prices based on the different features provided. The model’s R^2 score square is 0.2689189484498322, the MSE score is 217250.1581372743, and the RMSE score is 466.1010170953124.

There is a similar dataset in Kaggle that contains renting factors like rent prices, size, number of bedrooms, floor, area type, and so on in India. The researcher uses the dataset, builds 11 models(SVR, ANN, KNN, Ridge, Bayesian Ridge, Lasso, Random Forest Regressor, GradientBoostingRegressor, LGBM regressor, Cat Boost Regressor, XGB Regressor), and finds the optimal model with a minimum RMSE score. Then, the researcher combines the lowest five models and gets an RMSE score of 21203.046427392685 and an R^2 score of 0.79383058389995.

For this kind of dataset, the employed state-of-the-art method is neural network with multi-layer prediction. The conclusion

from existing work is similar to our finding that rental price is affected by factors like regions, the number of beds, size, electric vehicle charging access, wheelchair access, etc.

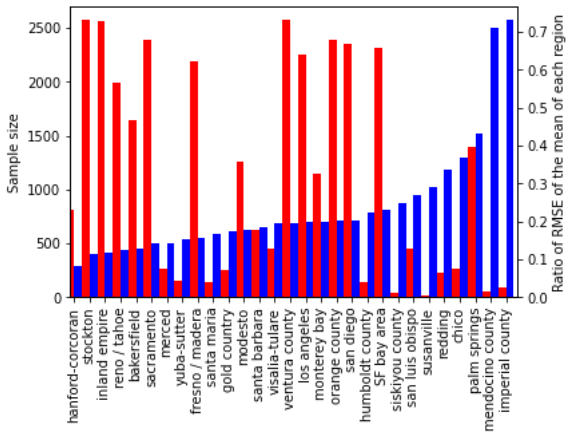
5. Results

In our study of models, we built a Ridge regression model, a linear regression model with features “id, region, price, type, sqfeet, beds, baths, cats_allowed, dogs_allowed, smoking_allowed, wheelchair access, electric_vehicle_charge, comes_furnished, laundry_options, and parking_options", a tf-idf vectorizer with the first 5000 most frequent words. Eventually, after comparing the results of different models, we decided to use the Ridge regression model since it has the lowest mean-squared-error compared to the rest.

For the linear Regression model, after fitting the model using our feature function that takes the value of and binds “id, region, price, type, sqfeet, beds, baths, cats_allowed, dogs_allowed, smoking_allowed,wheelchair access, electric_vehicle_charge, comes_furnished, laundry_options, and parking_options” into the feature vector, we imported the mean_squared_error function from sklearn.metrics and eventually came up with a mse value of roughly 628534. The square root of 628534 is about 793, which means that, using this linear regression model, the prediction we made, on average, is either about $793 higher or lower than the actual housing pricing, which is quite a big difference given that the housing prices for most housings are only around 1000-3000.

For the tf-idf vectorizer model, after fitting the model using the Tfidfvectorizer from sklearn.feature_extraction.text, we did the same mean_square_error computation and eventually came up with a mse value of roughly 9722866989757. This is the model that failed to give us any useful information since the value is too high to be comprehended. The reason why this might occur is because we did not deal with uppercase and lowercase and punctuations.

The third model, Ridge regression model, which is also the one we found the most accurate and useful one, gives us not just a simple mean squared error value, but a series of root mean squared error ratio of the mean of each area in California.



As the diagram shown above, the third model clearly shows that the fewer the number of data points exists, the higher the errors are, and vice versa. For example, we can see that for mendocino county, consisting of almost 2500 data points, only has about 0.01 (1%) of error. By this it means that the average difference between the actual price and predicted value for every data point is about 1% of the value of the mean housing price in mendocino county. On the other hand, if we look at the inland empire, while having roughly 650 data points, it has more than 0.40 (40%) of predicted error, meaning that the average difference between the actual price and the predicted price for every data point is more than 40% of the value of the mean housing price in inland empire.

The features we used in our Ridge model are “region, price, type, sqfeet, beds, baths, cats_allowed, dogs_allowed, smoking_allowed,wheelchair access, electric_vehicle_charge, comes_furnished, laundry_options, parking_options, and description”, and it turns out that this feature representation works the best for our model. For “region, price, type, sqfeet, beds, baths, cats_allowed, dogs_allowed, smoking_allowed,wheelchair access, electric_vehicle_charge, comes_furnished, laundry_options, and parking_options”, we used OneHotEncoder and transformed all these columns into corresponding vectors, and we also built a TfidfVectorizer, same to the failed TfidfVectorizer model, that transformed the description column into a tf-idf feature vector.

The parameter we used to initialize the OneHotEncoder is handle_unknown = ”ignore”, which means that when an unknown category is encountered during transform, the resulting one-hot encoded columns for this feature will be all zeros. In the inverse transform, an unknown category will be denoted as None. The parameter we used to initialize TfidfVectorizer is decode_error = “ignore”, which means to ignore if a byte sequence given to analyze contains characters not of the given “encoding” (another parameter of TfidfVectorizer, in our case we chose to use default value). The parameter we used for Ridge is alpha = 1.0, which is the Constant that multiplies the L2 term, controlling regularization strength.

Thus, given the property of our model that the more training data we have the better our predicted results would be, we are confident that our model can give accurate predicted results for the regions that have enough data points.

Reference

Boeing, G., & Waddell, P. (2016). New Insights into Rental Housing Markets across the United States: Web Scraping and Analyzing Craigslist Rental Listings. *Journal of Planning Education and Research*. https://doi.org/10.1177/0739456X16664789