The Three Data Scientists' Final Project Report

Project Idea

As we are in a recession and a lot of college students are going to be looking for places to live as they get their first jobs or look to move out of their parent's house we wanted to explore and ultimately predict the price of a house or apartment depending on the most looked-at features people prefer when looking for a place to live and have a reference of an areas average rent price or housing price.

Data

The data we used to create our machine learning models, exploratory data analysis, and visualizations were extracted from Zillow and apartments.com websites.

Creating Data Frames

Found in our GitHub we have our Python scrapper files and two Jupyter notebooks that pull data, create data frames, and show data visualizations. We had to do a bit of cleaning, dropping null values and columns that we did not find useful in our project and translating string values into numbered or ‘hot encoding’ some columns so they are easier to work with.

```
df1 = df1[df1['Rent'] != 0]
```

(only keeping rent values that arent 0)

```
79 / 100  (Excellent Transit)    15
74 / 100  (Excellent Transit)    12
75 / 100  (Excellent Transit)     8
100 / 100  (Rider's Paradise)     8
72 / 100  (Excellent Transit)     7
76 / 100  (Excellent Transit)     7
87 / 100  (Excellent Transit)     6
71 / 100  (Excellent Transit)     5
78 / 100  (Excellent Transit)     5
83 / 100  (Excellent Transit)     5
88 / 100  (Excellent Transit)     2
86 / 100  (Excellent Transit)     2
73 / 100  (Excellent Transit)     2
81 / 100  (Excellent Transit)     1
97 / 100  (Rider's Paradise)      1
77 / 100  (Excellent Transit)     1
84 / 100  (Excellent Transit)     1
90 / 100  (Rider's Paradise)      1
92 / 100  (Biker's Paradise)      1
Name: Transit Score, dtype: int64
```

(hot encode translatability)

```
79.0     15
74.0     12
75.0      8
100.0     8
72.0      7
76.0      7
87.0      6
71.0      5
78.0      5
83.0      5
88.0      2
86.0      2
73.0      2
81.0      1
97.0      1
77.0      1
84.0      1
90.0      1
92.0      1
Name: Transit Score, dtype: int64
```

After doing so we were able to set some parameters of our own and label homes as a specific type of home based on bedrooms and price.

```
def get_house_type(num_bedrooms, price):
    if num_bedrooms >= 4 and price >= 1000000:
        return "Luxury"
    elif 2 <= num_bedrooms <= 3 and 500000 <= price < 1000000:
        return "Residence"
    elif num_bedrooms <= 2 and price < 500000:
        return "Single Family"
    else:
        return "Other"
```

```
df2['Bedrooms'] = df2['Bedrooms'].replace("Studio", 1)
```

```
df2['Bedrooms'] = df2['Bedrooms'].astype(float)
```

For some places that didn't have features we wanted like transit score, we used the median data value to replace the null value.

```
median_score = df1['Transit Score'].median()
df1['Transit Score'].fillna(median_score, inplace=True)
```

```
median_score = df2['Transit Score'].median()
df2['Transit Score'].fillna(median_score, inplace=True)
```

We did so for both the Zillow data and for apartments.com data (df1 & df2) and then we merged them to create a single data table we can work with to do our exploratory data analysis and create models.

```
df1.columns
```

```
Index(['Address', 'Bedrooms', 'Bathrooms', 'Rent', 'Total Area',
       'Price per sqft', 'Type of house', 'Transit Score'],
      dtype='object')
```

```
df2.columns
```

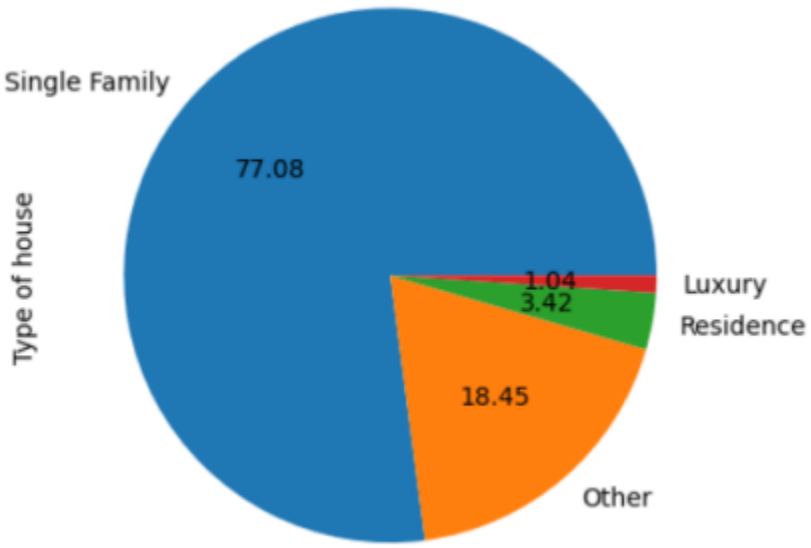
```
Index(['Address', 'Bedrooms', 'Bathrooms', 'Rent', 'Total Area',
       'Price per sqft', 'Transit Score', 'Type of house'],
      dtype='object')
```

```
merged_df.columns
```

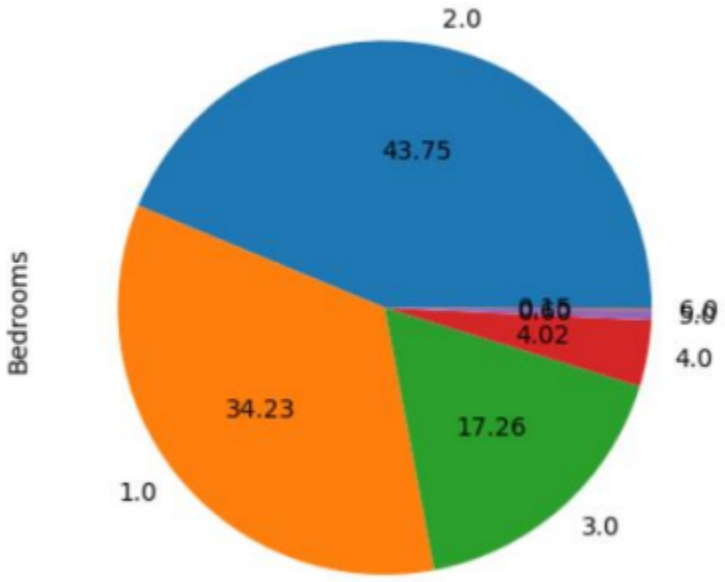
```
Index(['Address', 'Bedrooms', 'Bathrooms', 'Rent', 'Total Area',
       'Price per sqft', 'Type of house', 'Transit Score'],
      dtype='object')
```

Exploratory Data Analysis

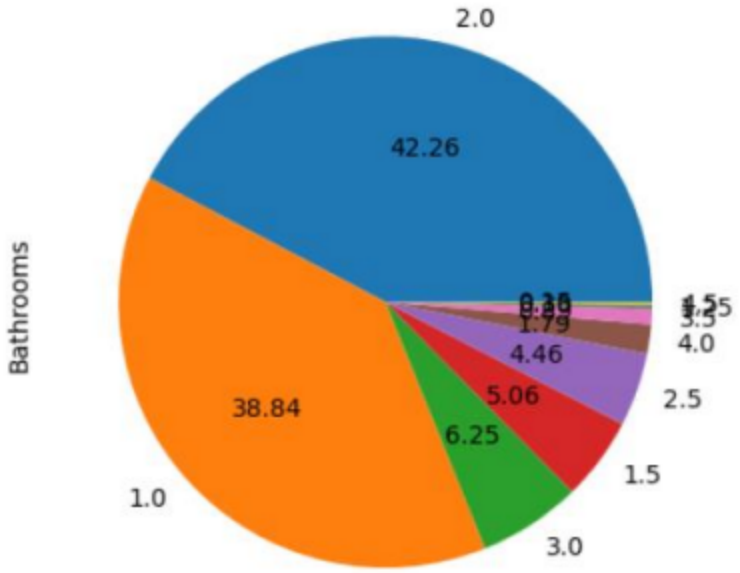
With our newly merged data frame, we created the following visualizations to show some big-picture data from the local UIC area zip code.



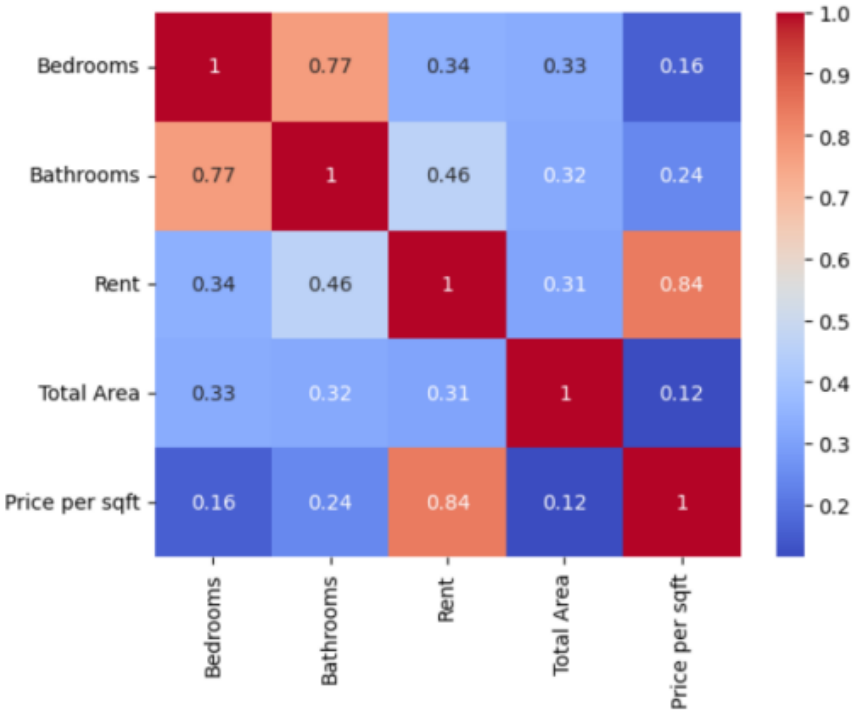
We found out that most of the types of houses are single-family homes or by our definition less than or equal to 2 bedrooms which makes sense since most homes and apartments are designed for college students or young adults with roommates.



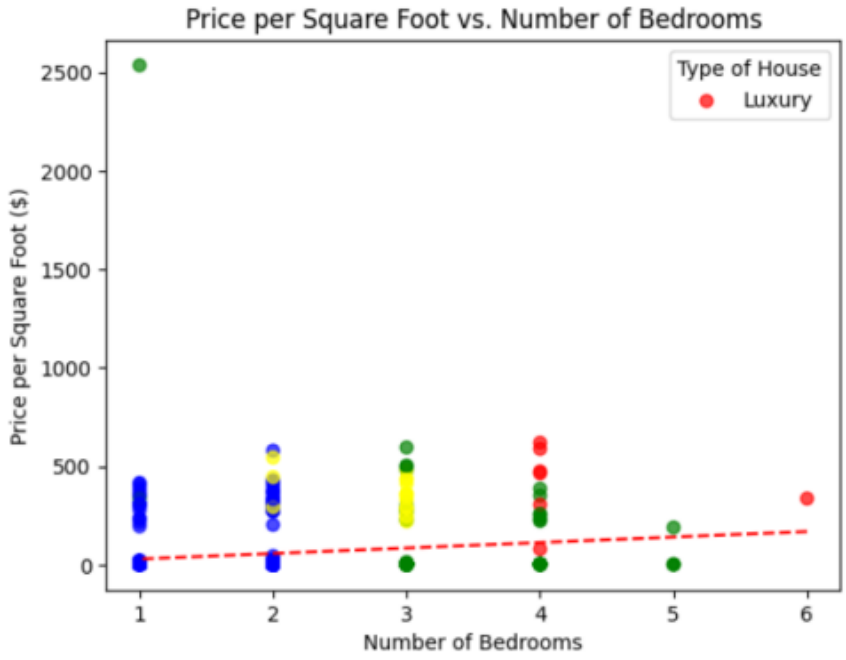
Most places also have 2 bathrooms which was surprising since most places that are 2 bedrooms typically would only have 1 bathroom.



With that being said we wanted to see if there was any relationship between rent/price so we created a heatmap to show that and found that the most correlation was between rent and total area/price per sqft which makes sense since bigger places usually have more rooms or just more space to relax in so they would be more expensive.



We also created a linear plot that showed an increase in price per sqft as rooms increased.



ML Models

The Y feature (Rent) was separated as y and the rest of the variables as X. The address variable was dropped as it has no relation.

The data was split into train and test set with a ratio of 80/20 respectively. To normalize data standard scalar on Scikit Learn is used.

The train set is fitted and transformed, and the test set is transformed.

Training and Testing

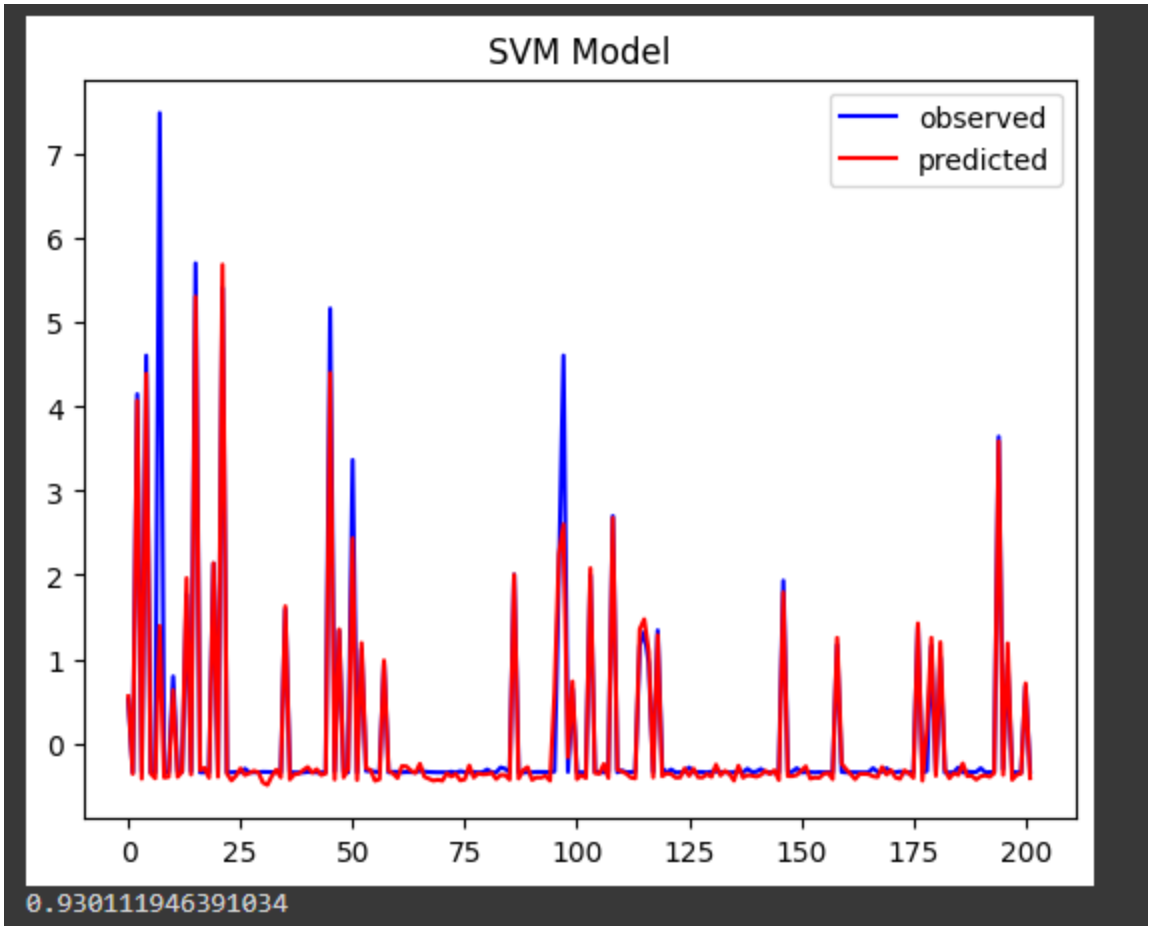
The linear regression instance was initialized, and train data was fit and evaluated on the test dataset.

Results:

Following were the results after evaluation:

```
R-squared: 0.801
MSE: 22129744547.294
RMSE: 148760.696
MAE: 43989.748
```

We also created a Support Vector regression model that was scaled using Standard Scalar and split 70/30 and we ended up with a 93% accuracy level when predicting price using bedrooms, bathrooms, total area,price per sqft, and transit score.



Explanation: The point of this SVM regression model was to try and take hyperparameters that a person might want when looking for a home or apartment and try and predict a price given those parameters. This would come in handy when looking over a specific zip code and loading in that data to the model and the person could get a pretty accurate price or see what price on average is the place they are looking at so they can negotiate better with the landlord. I believe if we added in hyperparameters like if the location is considered a luxury or a house vs an apartment the accuracy could be increased since those parameters are more volatile depending on the state of the market. This model wouldn't be very good overall over large scales to guess prices in states or in the country since the average cost of living changes and a model trained on

Wyoming data wouldn't be very accurate in California, so going through and training and testing in specific towns, county, or zip codes would share this kind of accuracy we produced.

Reflection

Hardest Part

The hardest part of this project was to extract data from the websites, clean, and format the data so it was easy to work with and insert into our models and get that clear picture with the visualizations. We learned initially from Prof Ziebart when we started the class that data science is 80% extracting and cleaning data versus actually analyzing and creating concrete results from it and that proved true.

Results

We definitely learned that this type of research and experimentation is what is behind the housing market and how pricing is calculated based on those parameters we have mentioned in the project, so when contractors bulldoze and clear old houses and develop complexes of luxury apartments in an area we see that correlation that brings up the average cost of sqft in the area. If we were to continue on this project we could definitely dive into gentrification and use future developments plans in the area to analyze what improvements or changes affect housing prices the most like, bike lanes, parks, or adding single-family homes in the area over apartment complexes and also take into account liveability and safety of the area.