Notebook

March 28, 2021

1 Part 0: Background

Let's try to understand the background of our dataset before diving into a full-scale analysis.

1.1 Question 0

1.1.1 Part 1

Given that this data compiled by the CCAO (as opposed to other assessor's offices), find a column that would most likely only be collected in Cook County. What does this column represent?

The 'Town and Neighbourhood' column would be unique to Cook County because other Counties would have different neighborhoods and towns.

1.1.2 Part 2

Name a feature that isn't listed in this dataset but may be useful for predicting sales values. What insights could this feature provide? How might it increase or decrease a home's sales value?

Pool sqft - size of the swimming pool in square feet. defaults to 0 if there is no swimming pool. It might increase the home's value if the swimming pool is built well.

1.2 Question 1

1.2.1 Part 1

Identify one issue with the visualization above and briefly describe one way to overcome it. You may also want to try running `training_data['Sale Price'].describe()` in a different cell to see some specific summary statistics on the distribution of the target variable. Make sure to delete the cell afterwards as the autograder may not work otherwise.

[9]:
```
training_data['Sale Price'].describe()
```

[9]:
```
count    2.047920e+05
mean     2.451646e+05
std      3.628694e+05
min      1.000000e+00
25%      4.520000e+04
50%      1.750000e+05
75%      3.120000e+05
max      7.100000e+07
```

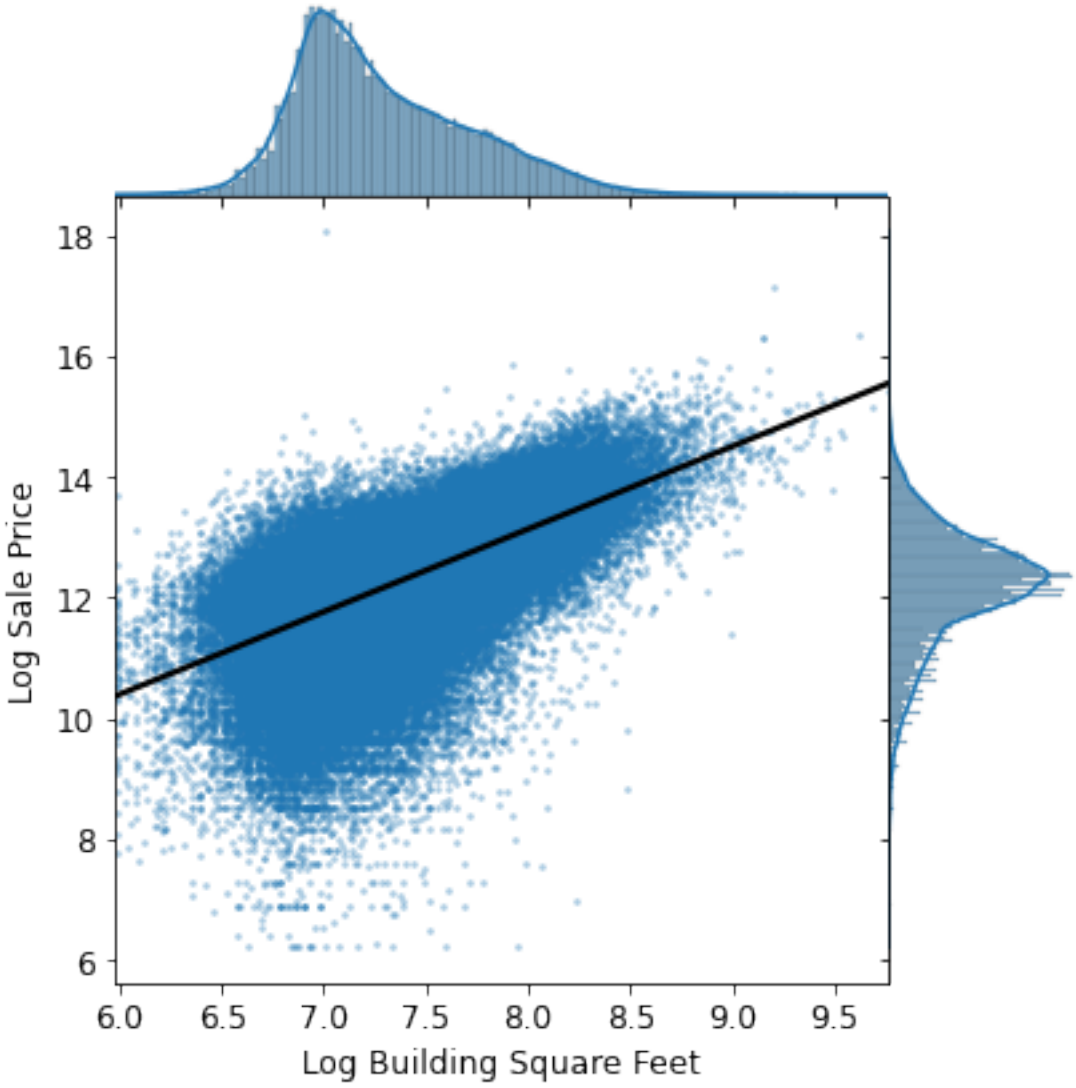
```
Name: Sale Price, dtype: float64
```

Well the above visualization is uninterperatable. We could take the log of the sale price to overcome it?

1.2.2 Part 3

As shown below, we created a joint plot with `Log Building Square Feet` on the x-axis, and `Log Sale Price` on the y-axis. In addition, we fit a simple linear regression line through the bivariate scatter plot in the middle.

Based on the following plot, does there exist a correlation between `Log Sale Price` and `Log Building Square Feet`? Would `Log Building Square Feet` make a good candidate as one of the features for our model?



Yes they are kind of related, I would not say that it would make a good candidate for our model

but it would be a decent candidate to use among others.

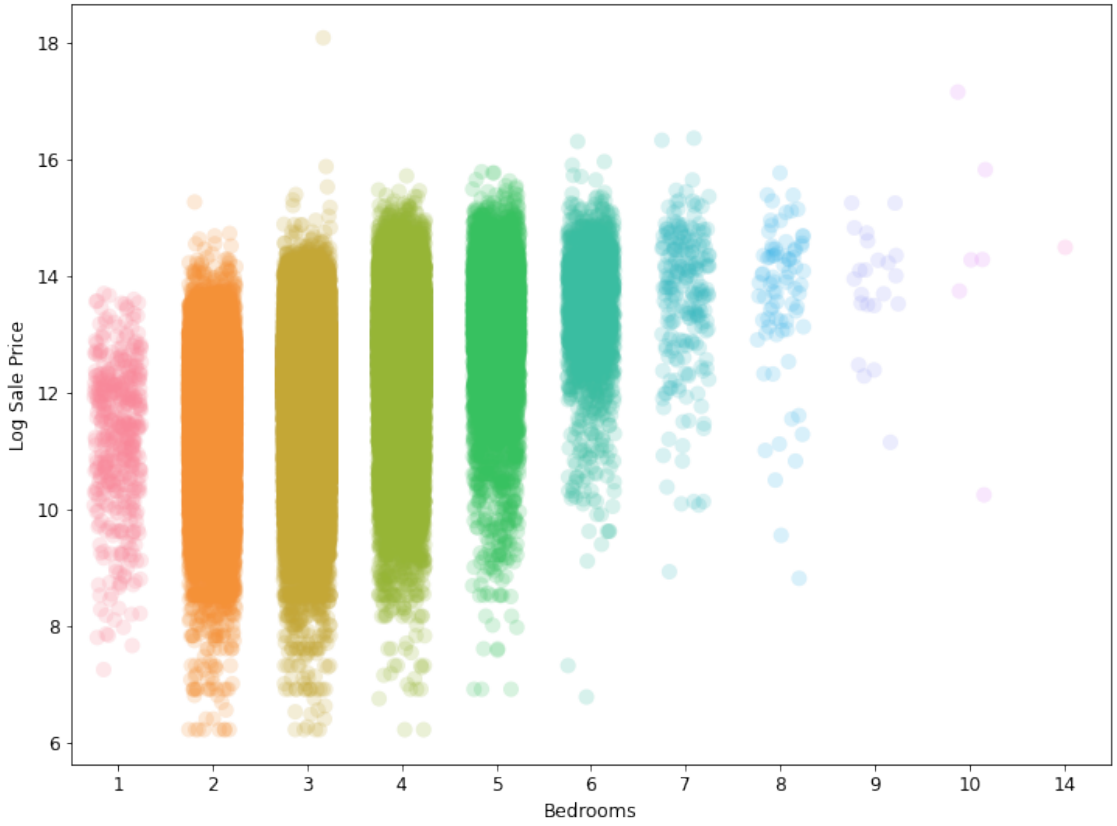
1.2.3 Part 3

Create a visualization that clearly and succintly shows if there exists an association between `Bedrooms` and `Log Sale Price`. A good visualization should satisfy the following requirements: - It should avoid overplotting. - It should have clearly labeled axes and succinct title. - It should convey the strength of the correlation between the sale price and the number of rooms.

Hint: A direct scatter plot of the sale price against the number of rooms for all of the households in our training data might risk overplotting.

[32]:
```
sns.stripplot(data=training_data, x='Bedrooms', y='Log Sale Price', jitter=0.
↪25, alpha=0.2, size=10)
```

[32]:
```
<AxesSubplot:xlabel='Bedrooms', ylabel='Log Sale Price'>
```



1.2.4 Part 3

It looks a lot better now than before, right? Based on the plot above, what can be said about the relationship between the houses' `Log Sale Price` and their neighborhoods?

The `Log Sale Price` is somewhat dependant on the neighborhood because some neighborhood codes have a median value above or below the median price implying that those neighborhoods have a higher or lower price respectively.

1.2.5 Part 2

Without running any calculation or code, complete the following statement by filling in the blank with one of the comparators below:

$$\geq$$
$$\leq$$
$$=$$

Suppose we quantify the loss on our linear models using MSE (Mean Squared Error). Consider the training loss of the 1st model and the training loss of the 2nd model. We are guaranteed that:

$$\text{Training Loss of the 1st Model} \_\_\_\_\_ \text{Training Loss of the 2nd Model}$$

$$\geq$$

1.2.6 Part 6

Let's compare the actual parameters (θ_0 and θ_1) from both of our models. As a quick reminder,

for the 1st model,

$$\text{Log Sale Price} = \theta_0 + \theta_1 \cdot (\text{Bedrooms})$$

for the 2nd model,

$$\text{Log Sale Price} = \theta_0 + \theta_1 \cdot (\text{Bedrooms}) + \theta_2 \cdot (\text{Log Building Square Feet})$$

Run the following cell and compare the values of θ_1 from both models. Why does θ_1 change from positive to negative when we introduce an additional feature in our 2nd model?

[64]:
```
# Parameters from 1st model
theta0_m1 = linear_model_m1.intercept_
theta1_m1 = linear_model_m1.coef_[0]

# Parameters from 2nd model
theta0_m2 = linear_model_m2.intercept_
theta1_m2, theta2_m2 = linear_model_m2.coef_

print("1st Model\n 0: {}\n 1: {}".format(theta0_m1, theta1_m1))
print("2nd Model\n 0: {}\n 1: {}\n 2: {}".format(theta0_m2, theta1_m2,␣
 ↪theta2_m2))
```

```
1st Model
 0: 10.571725401040084
 1: 0.4969197463141442
2nd Model
 0: 1.9339633173823714
 1: -0.030647249803554506
 2: 1.4170991378689641
```

If you have too many bedrooms in a small building it will seem cramped which may affect the prices negatively.

1.2.7 Part 7

Another way of understanding the performance (and appropriateness) of a model is through a residual plot.

In the cell below, use `plt.scatter` to plot the predicted `Log Sale Price` from **only the 2nd model** against the original `Log Sale Price` for the test data. You should also ensure that the dot size and opacity in the scatter plot are set appropriately to reduce the impact of overplotting.

[65]:
```
sns.scatterplot(x=y_predicted_m2, y=y_test_m2, alpha=0.2, s=5)
```

[65]: <AxesSubplot:>