Overview

I think the best topic for me to cover for my first project is simply an overview of my thoughts, assumptions, and reasoning for why I approached it the way that I did. This is helpful for me because sometimes I find myself doing things that aren’t always the clearest, even to myself, and this forces me to verbalize those things. I think it would also help students who follow me because I found myself wishing, a number of times, that I could really get inside the mind of student’s whose work I read over; though there were some explanations in their projects, it was more a description of what they were doing, not a rational as to why.

Initial EDA

Initially cleaning and analyzing the data was pretty straight forward and was pretty explicitly covered in the lessons that went over EDA. Lots of:

|  |  |
| --- | --- |
| * .describe | * .loc |
| * .info | * .head |
| * .value\_counts | * .nunique |
| * .iloc | * .isna |

These tactics mainly helped me to deal with outliers, null values, and incorrect data types. I also transformed the categorical data to dummy variables. Let me say at the outset: dummy variables are a little hard for me to wrap my mind around. Not necessarily the concept of them—that’s clear enough—but when it comes to interpreting them and playing with them, I found myself asking a lot of questions that I found the curriculum unable to answer. I’ll get to some of these questions a little later when we get to the model results.

Finding the Best Variables

After cleaning the data my objective was to see which independent variables were strongly correlated with the target (price) and put those variables aside to use in the OLS/Least Squares model. Then I wanted to see if any of the variables in that subset were correlated with each other, to avoid multicollinearity. If any of them were correlated, I was going to drop (or not include in the final model) whichever of the two (or more) variables was less correlated with the target.

At this point, I think it would have been appropriate to see if there was any way to feature engineer (which I take to simply mean creating new predictive variables out of the variables you already have) some of the variables we were given. I very much wanted to use either the zip code or the latitude/longitude, because I know that location has enormous predictive power in the real-world real estate market—but, much to my chagrin, I found myself unable to come up with anything that I was confident would pass a smell test. In hindsight I think there are a couple things I could have come up with, but at the time I was drawing blanks and feeling a little pressure due to the timeline I was on, so I moved on to something else.

‘Tis the Season

That thing was seasons. It seemed common sensical to me that the time of year would have an effect on the real estate market. I mean, companies turn calendar years into fiscal years by breaking them up into four fiscal quarters to help them strategize on how to run their business depending on what time of year it is. After investigating, I found out that the seasons did indeed have an effect on the real estate market in King County, but not in the way I expected. Here’s what I did.

When you turn a date into a date-time object, you will have access to functionality that you wouldn’t have had if the date were stored as, say, a string. Using a built-in method, I was able to easily extract the month in which the sale of the home took place. After I had the months, I broke them into the four seasons. After *that*, I was able to put those into their own mini data frame with their value counts so that I ended up with two columns and four rows: one column was the season’s name, and the other was the number of times a house was sold in that season.

I found my hunch that homes sold better in the warmer months to be true; summer and spring were almost tied, and outstripped the next season, Fall, by about 1,000 values. What was curious was that, when I plotted the seasons against the price, our target, I found that the season had little effect on the home’s price. My hypothesis was that the seasons would affect the prices by representing their own small buyer’s or seller’s markets: since in the warm months there was more demand, I actually thought the housing prices would be lower to compete with other sellers, and that in the colder months the prices would be higher since there was less stock to choose from. But I found that sales were almost flat across the seasons with the exception of a few high outliers during the Fall.

Experimenting with OLS

There weren’t a whole lot of other great candidates for feature engineering: I had already given up on location at this time (I was just going to include it in the model without doing much to it), there were some variables that I dropped like “view” and “waterfront” because they were Booleans that had very low positive value and wouldn’t have affected the rest of the data that much; and then there were all the square-footage variables—most were correlated with each other or were practically built upon each other. For example, if I’m not mistaken, I believe “sqft\_above” and “sqft\_basement” added up to “sqft\_living.” Then there was the square footage of the 15 closest neighbors, but those were also correlated with the square footage of the home in question, and I was afraid that if I used those I would run into a multicollinearity problem.

So I began to plug things into the OLS model. At first I was only getting R^2 scores in the high 50’s to low 60’s—then I remembered scaling & normalization. To be frank once more, this was also a topic I didn’t completely feel comfortable with. It was similar to the dummies—I understood the process but not too much of the intuition. I didn’t really know when to use which technique. I ended up using log transformation because I did know that you were supposed to use that if your residuals were skewed, and I knew from earlier exploration that both my target variable, and my strongest independent variable, “sqft\_living,” were both skewed.

I log transformed all the continuous variables and the target. My R^2 score shot up about 10 points to just over .71. I was satisfied with this. I knew I could probably go back to fine tune certain things, or maybe do some more feature engineering, but I had already spent almost two weeks on the project, and it was time to move on.

Model Interpretation

To this day (but not to all days in the future!) I don’t fully understand how to interpret my coefficients in the OLS results, especially with the dummy variables. Take grade, for example. All of the coefficients are negative, which struck me as strange, because, according to common sense, and also according to when I was analyzing grade before I broke it up into dummy variables, the higher the grade, the higher the home price, in general. Why were grade’s coefficients negative? And how do I put them into words? Something along the lines of “This particular dummy variable, keeping in mind it is only one of several other dummy variables that refer to the same overall variable, affects the home price in \_\_\_\_\_ magnitude, due to its coefficient, BUT, only when it applies to the observation in question. That is, if the observation had a grade of 9, but we are looking at grade #7’s coefficient, the latter has no effect on the former.”

My inclination, then, is to say (for example), “Grade #7 affects the home price negatively, by its coefficient, for each degree of *increase* the target has, but only if the home was given a grade of 7.” My other question was how to interpret the column that I had to drop in order to even be able to use dummy variables in the first place. I dropped grade #11’s column. Which means if all of the grade dummies have zeroes for values, then that particular observation was rated at grade 11. But how do I know the magnitude grade 11 has on the target if I don’t have a coefficient for it?

Now What?

There are still a few unanswered questions I have with regard to multiple linear regression using Statsmodels. I double checked the soundness of my model using train test split, mean squared error, mean absolute error, cross validation, a comparison of the R^2 scores for both the training and testing data, and I plotted my estimated y-values (y\_hat) against my real y-values for both my training and testing data. If your model is sound, this last test should end up following a 45 degree, or 1-to-1, trend. My real and estimated values had a very strong correlation for both sets of data, so all signs are pointing toward the model being a good one.

However, since I am a student, I still need to find answers to the questions I have. So I will do a follow-up blog post in which I will hopefully be able to present these answers. One thing I will say to my own credit, is that I wouldn’t have had the foggiest of how to do *any* of this just a month ago, which is a testament to how far you can come with hard work. And there’s still a long way to go.