Ames Housing Project

Justin Fischer
GA DSI-11-DEN

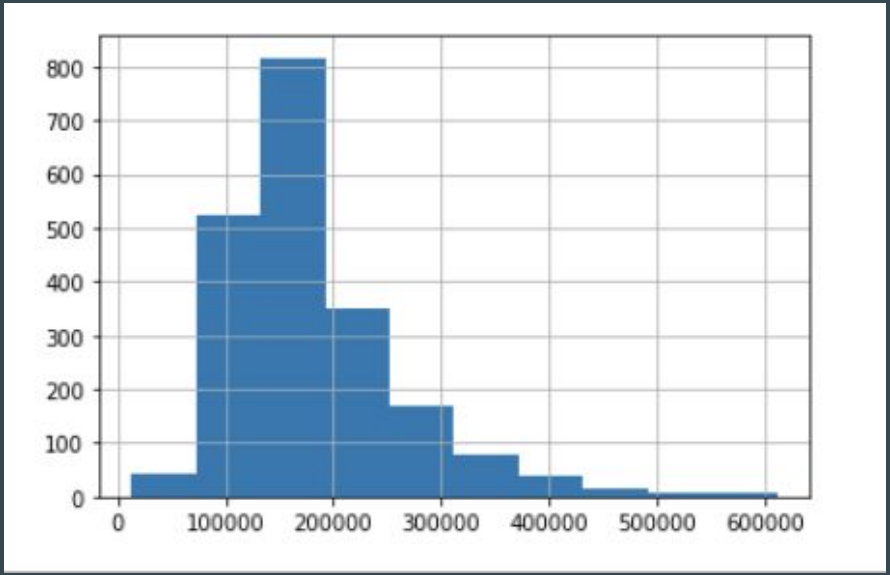
The Data

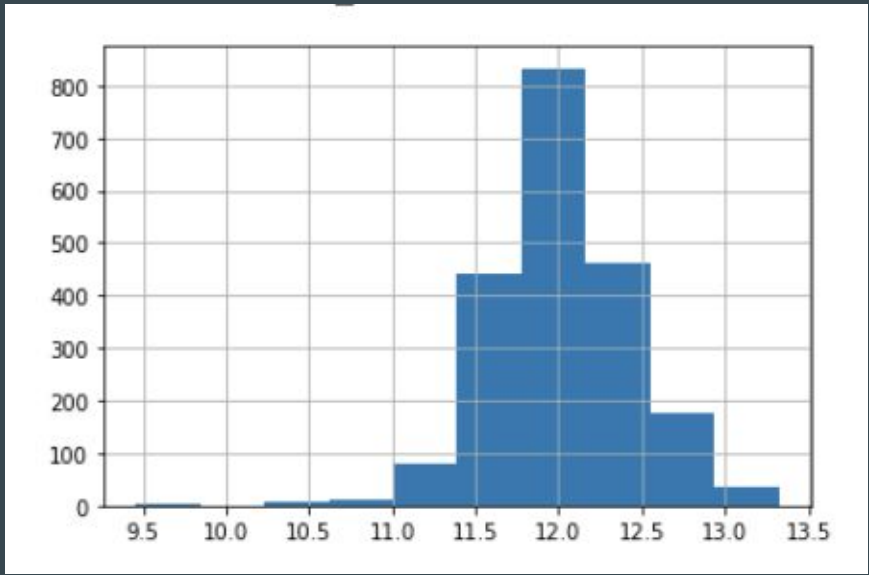
- Presented with 2 datasets.
  - Training - 2050 entries
  - Testing - 878 entries
- These datasets contain information on the houses sold in Ames, Iowa between 1/2006-12/2010
- Information includes:
  - Size of house/property
  - Number of bathrooms/bedrooms
  - Location
  - Condition/quality of various features
  - Sale date/type
  - Type of house
  - Various other features and attributes (basement, pool, fence, etc)

Methodology

- Started simple, went more complex
- First model
  - Simple linear regression with just square footage and number of bedrooms
- Added more features
  - Such as..
- Took the natural log of the sale prices to normalize it
- Rotated features
  - Neighborhood, Sale Type, House Type

Price Distributions





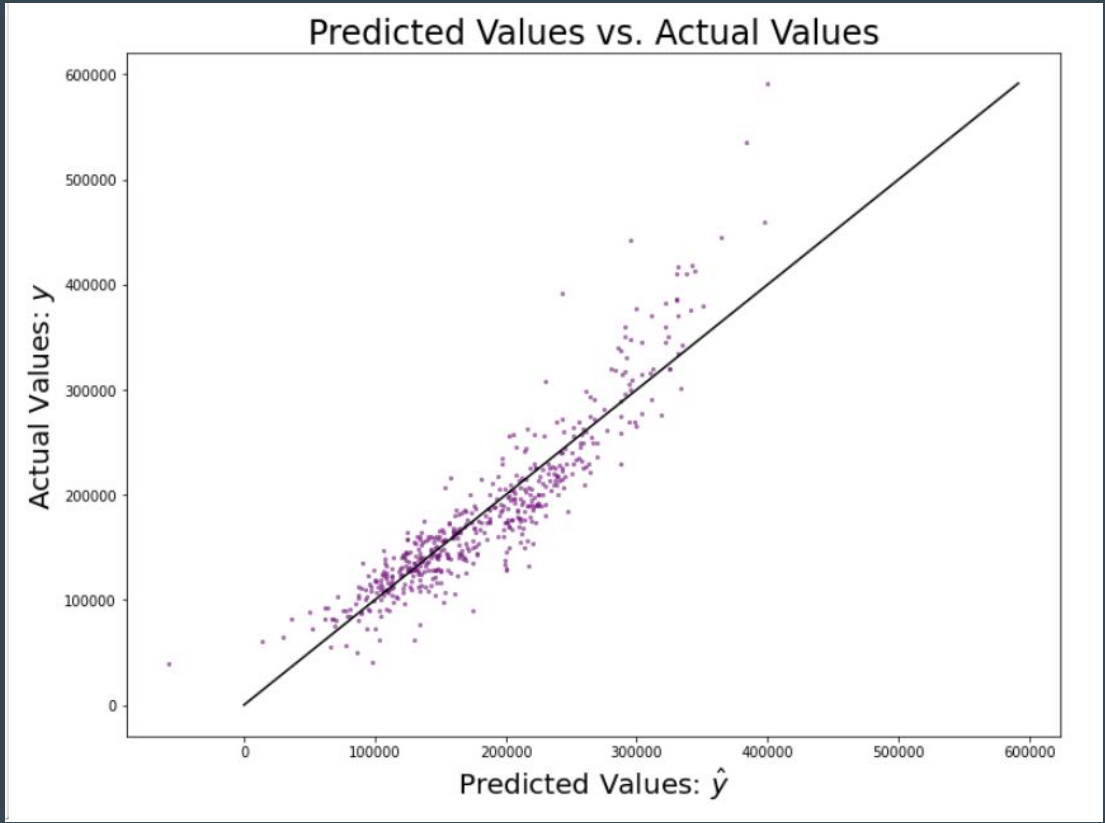
Simple First Regression

- Just looked at number of bedrooms and total square feet
- Root Mean Squared Error (RMSE) - $44,598

Select features

- Total square feet, Overall Quality (1-10), Exterior Quality (0-5), Kitchen Quality, Basement Quality, Garage Area, Number Bathrooms, Basement Square feet, Year Remodeled/Added/Built, Total Rooms Above Ground, Lot Area, Functional (0-7), Total Space Above Ground
- Training data RMSE - $31,117
- Then normalized by taking the natural log of sale price:
- RMSE - $28,640

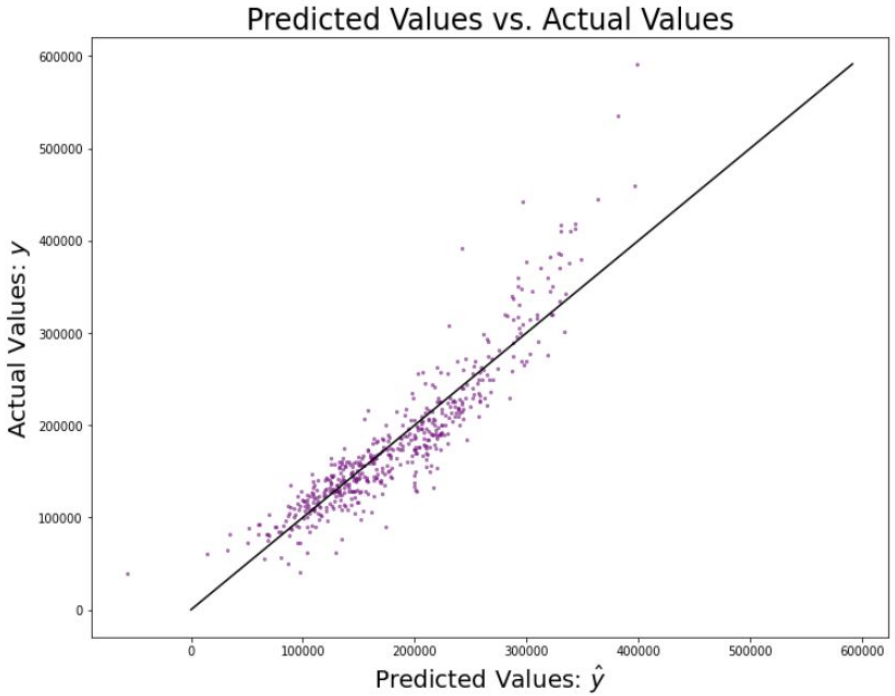
Select Features Errors

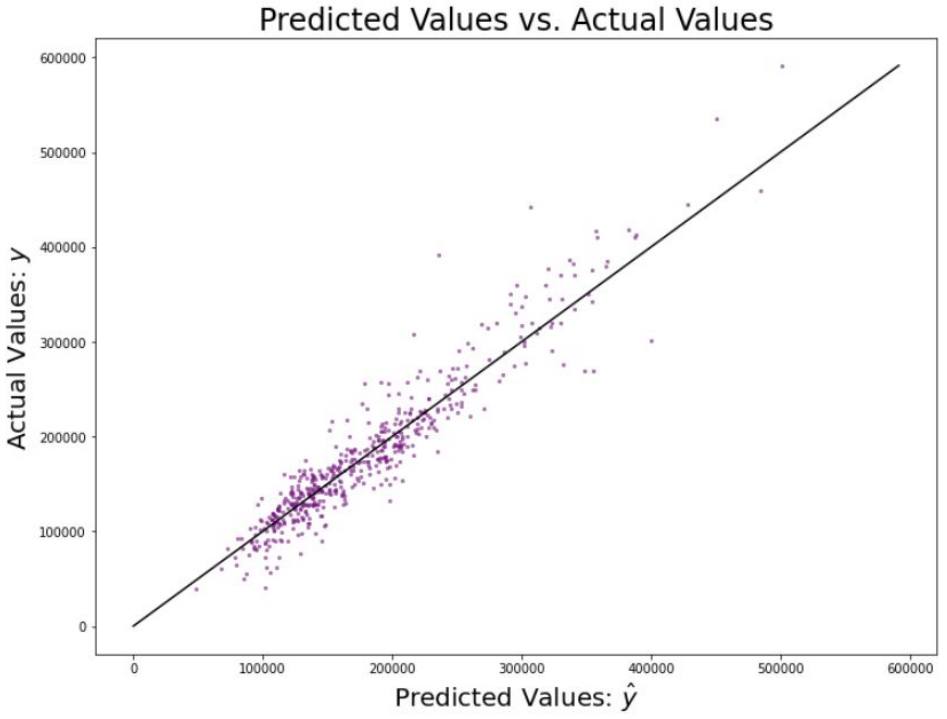


The Ridge

- Started to work on weighting some of the features using a ridge regression
- Sale price: $36,240
- Log Sale Price: $28,252
- This is where normalizing really started to pay off and errors dropped

No log v. log errors



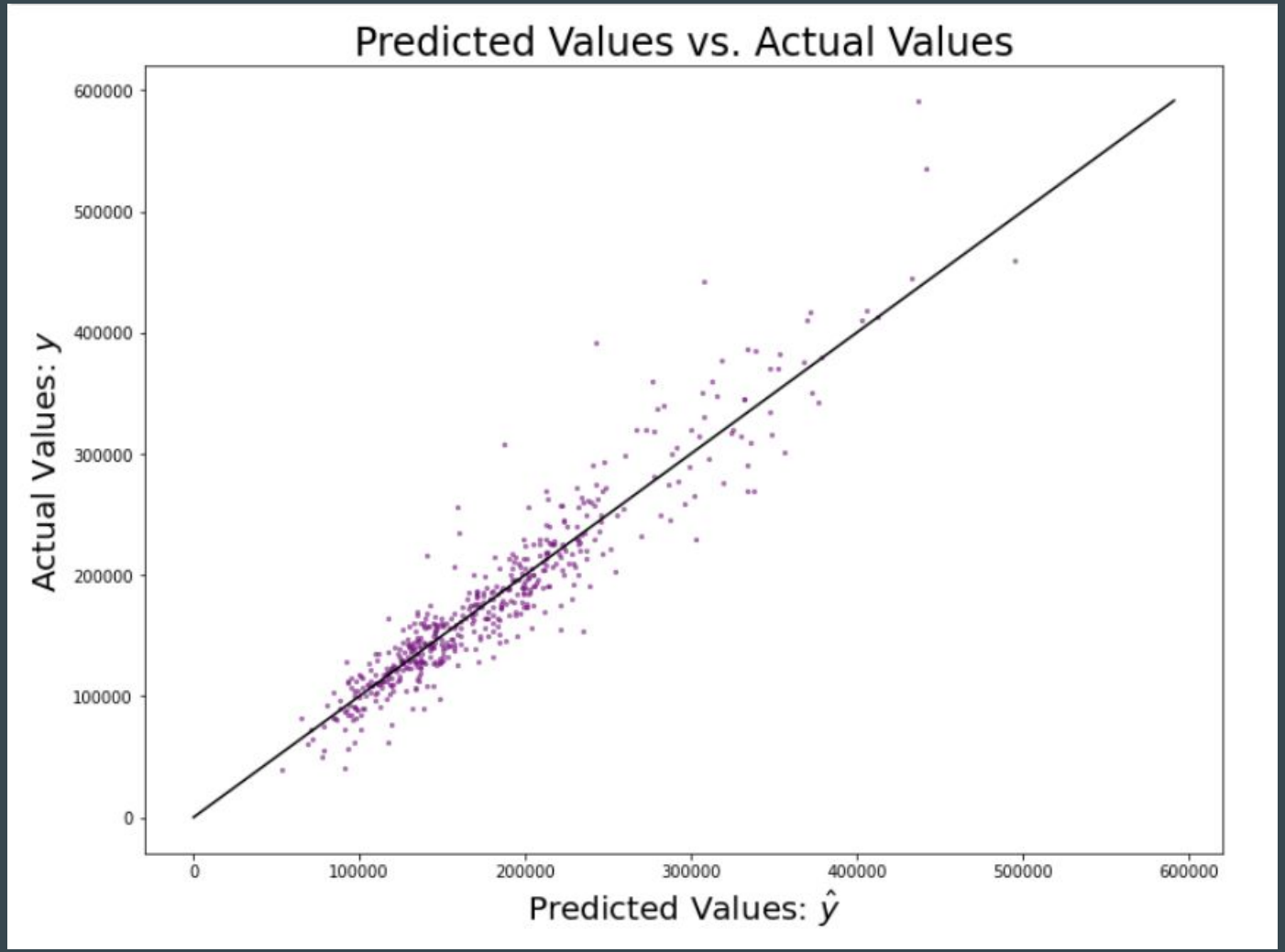


Categorizing Features

- Took a number of the features and created “dummy variables”
- Neighborhood
- Sale Type
- House Type
- Dwelling Type
- Pool

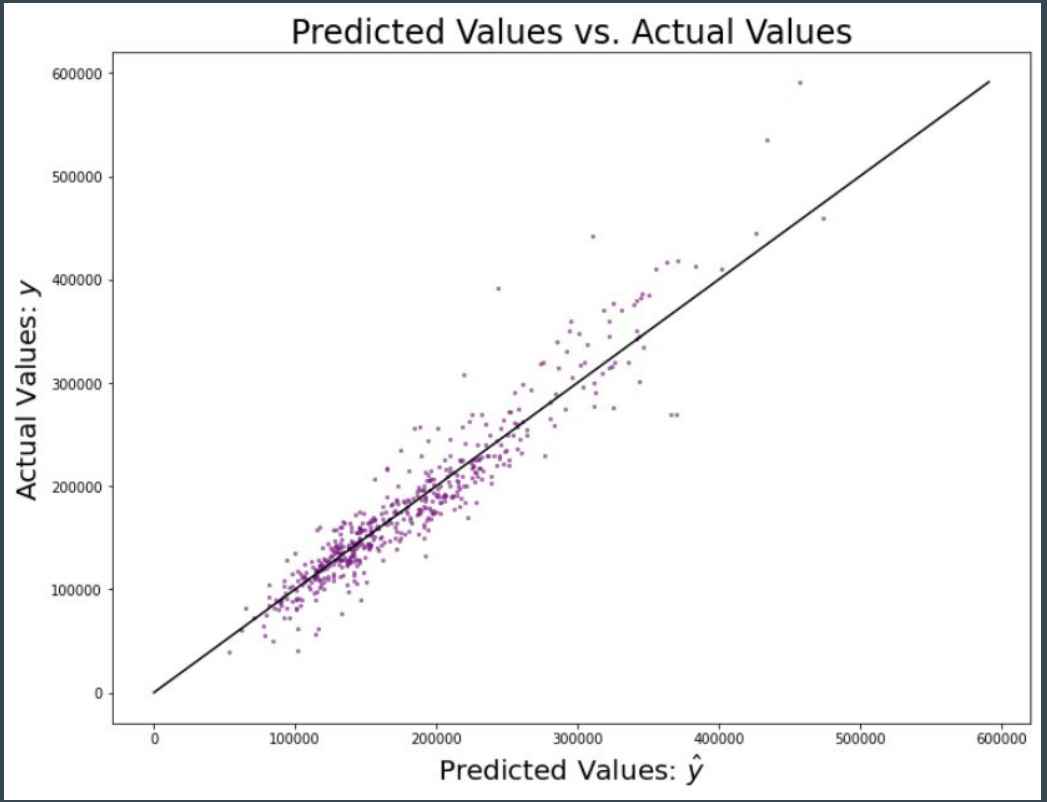
Neighborhood

- $28,252
- Basically the same as without neighborhood
- Not factoring in location seems counterintuitive, but that could be more on Ames than anything


Predicted Values vs. Actual Values
Actual Values: y
Predicted Values: $\hat{y}$
0
100000
200000
300000
400000
500000
600000

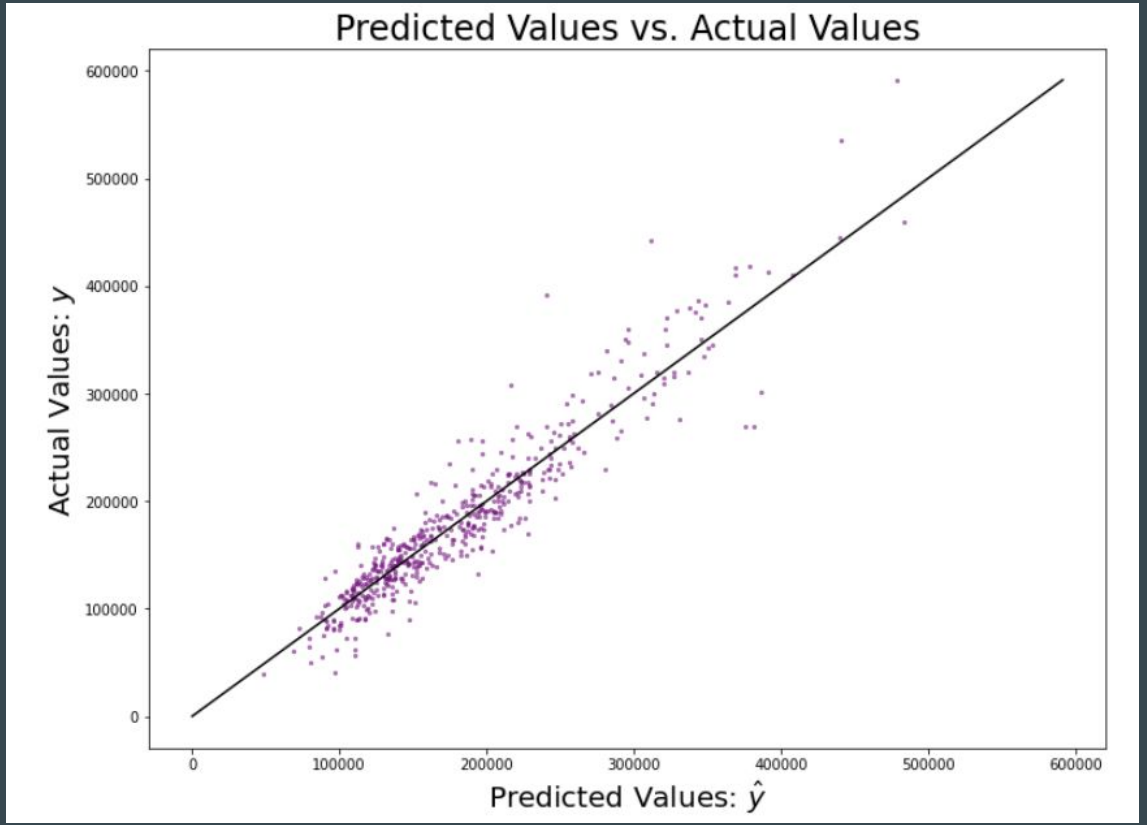
Class

- Types/ages of buildings
- $25,638



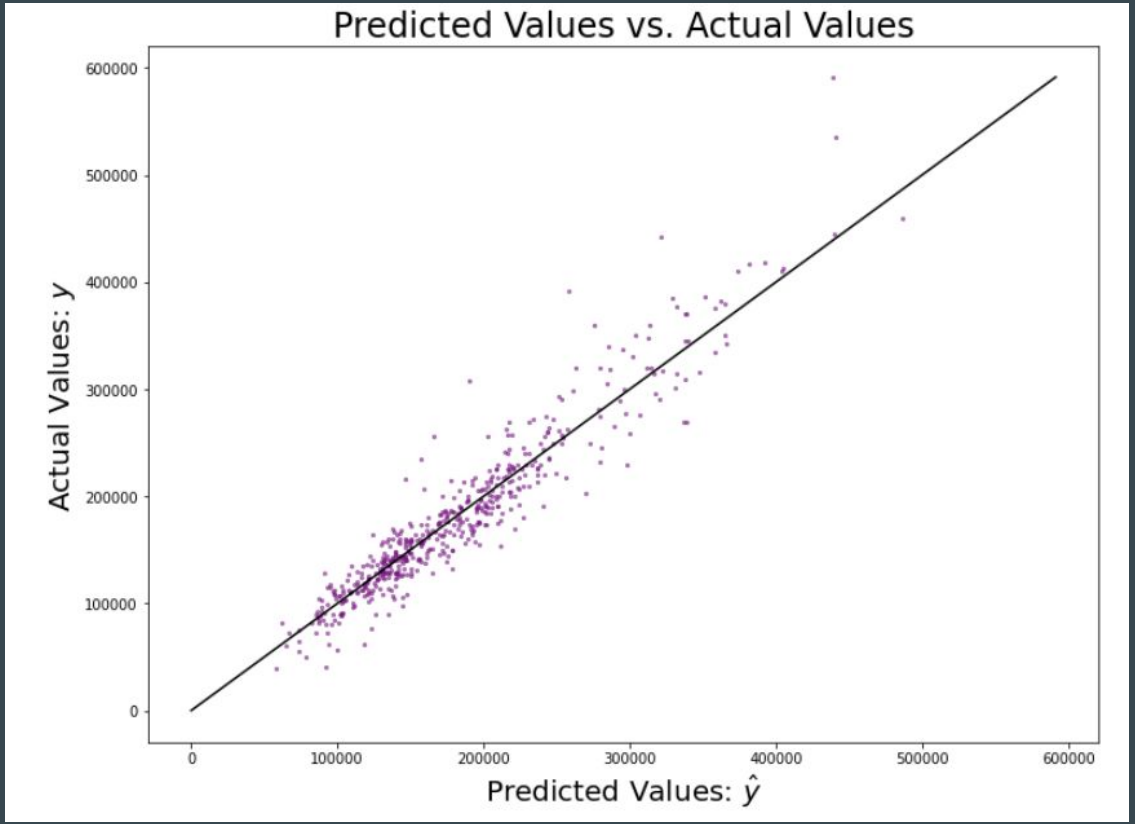
Style

- Training RMSE - $25,691
- This one was higher than class, so didn't submit it. Not as good a classifier



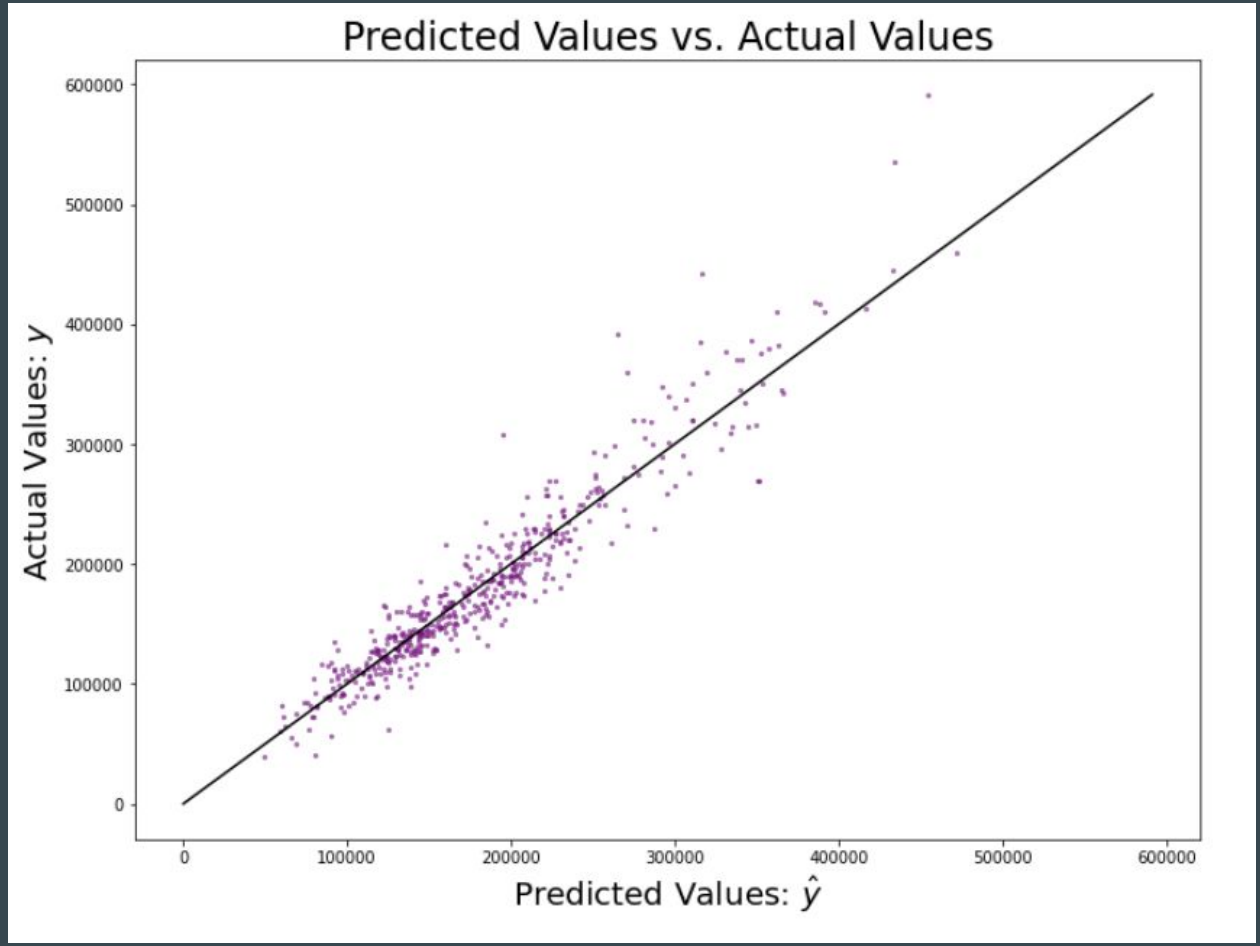
Class and Neighborhood

- Neighborhoods tend to be slightly different, factoring in the class of the building may help
- $24,505



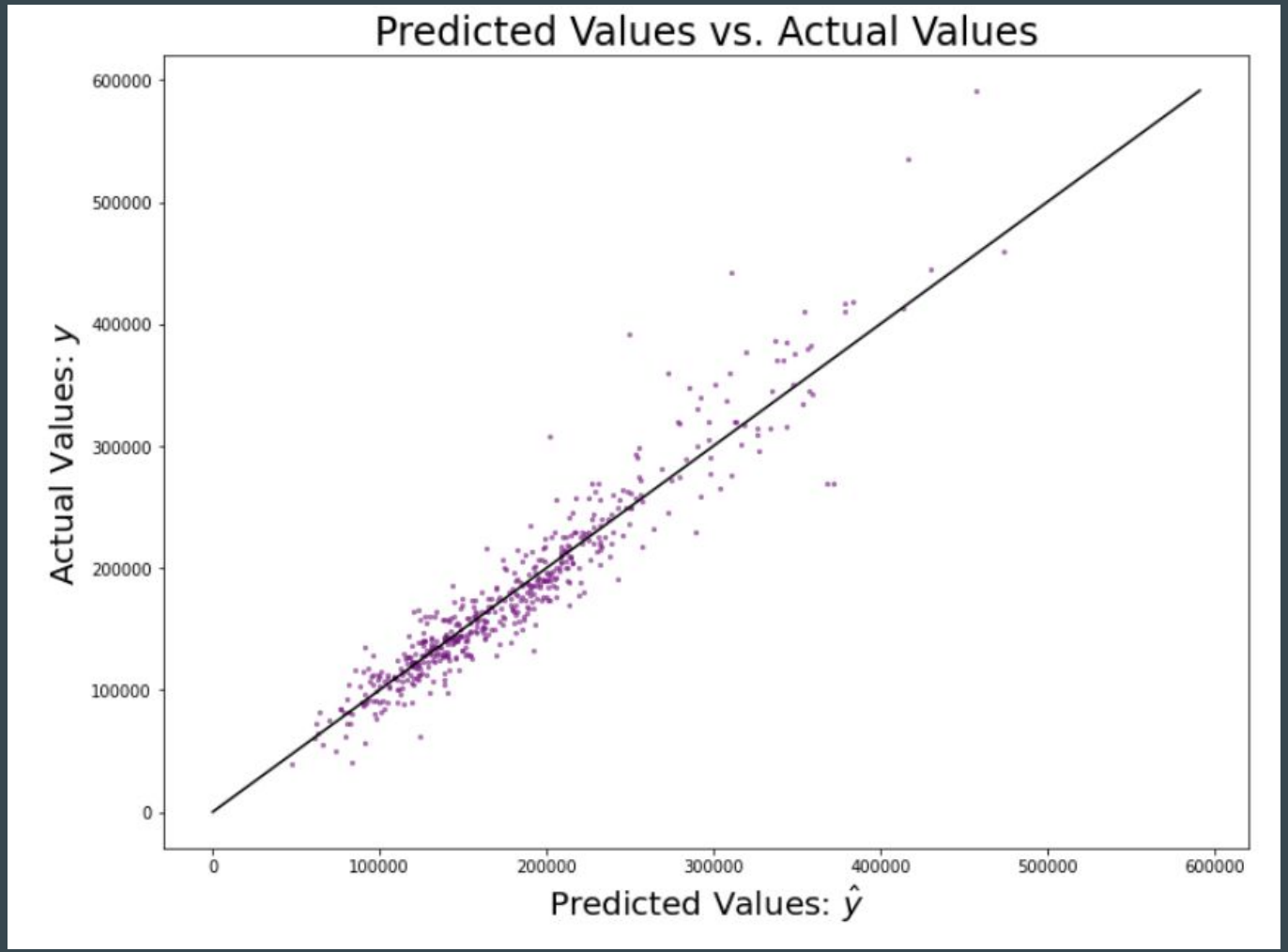
Many Features

- Pulled most of the features of the dataset that were quantifiable and looked like good classifiers.
- $23,732
- Not quite everything, probably room to add in data about proximity to potential noise


Predicted Values vs. Actual Values
Actual Values: y
Predicted Values: $\hat{y}$
0
100000
200000
300000
400000
500000
600000

Sale Type

- How a home is paid for could ultimately affect the price
- $28,258
- This is probably more unpredictable because of variance and different parties financial status
- In depth terms or rates weren’t shown


Predicted Values vs. Actual Values
Actual Values: y
Predicted Values: $\hat{y}$
0
100000
200000
300000
400000
500000
600000

Conclusions

- Mean sale price from training set is $181,461
- A 13% error would make me worried about trying to model this using the current setup

	RMSE	Percent
Simple	$44,598	24.58%
Select Features	$31,117	17.15%
Select Features - log	$28,640	15.78%
Ridge	$36,240	19.97%
Ridge - log	$28,252	15.57%
Neighborhood	$28,252	15.57%
Class	$25,638	14.13%
Style	$25,691	14.16%
Class+Neighborhood	$24,505	13.50%
All Features	$23,732	13.08%
Sale Type	$28,258	15.57%

Post-Conclusion

- However, model useless now since data is from 06-10. Need to account for the financial crisis in 2008.
- Need to rebuild it with a few other factors to be applicable to everywhere.
  - Economy
  - Supply/Demand in the region
  - Trend of people moving into/out of the region

Next Steps

- Make neighborhood work more.
  - Would need more details of each neighborhood and the demographics
- Figuring out what people want in the area to properly weigh everything
-