More Pictures of Cats Please!
Lukas Froehlich
Waylin Wang

Project 3

Data Acquisition

We decided to get our data only from slocountyhomes.com/newlistex.php. We wrote a script that scrapes the website every night and generates a csv file of house listings. Over the two weeks we were able to collect an additional 200 listings. We used the python library beautifulsoup to scrape the html documents to get the listings. One important task was merging the different snapshots of the website. This was done by iterating through the listings from oldest to newest and keeping a dictionary of the newest listings.

Data Cleaning

There wasn't much data cleaning to do for this data set since we did not have many features from the site that we decided to use. There were some cleaning involved with the city column to get rid of white spaces and getting rid of the comma in the price and square footage of the house. In some cases there were missing values for the city or street name and this was resolved by removing those observations.

Additional Features

The given features consisted of: MLSNumber, Street, City, Price, BR, Bath and Footage. These features give us a solid representation of the house but we felt that we could add additional features that could provide more interesting results. We first used the street and city and plugged them into Google maps' geocode API to get the zip code of the homes. Next we broke the prices into categories by binning. We did this to see if we could roughly estimate which financial category a house belonged to given its characteristics.

Classifiers

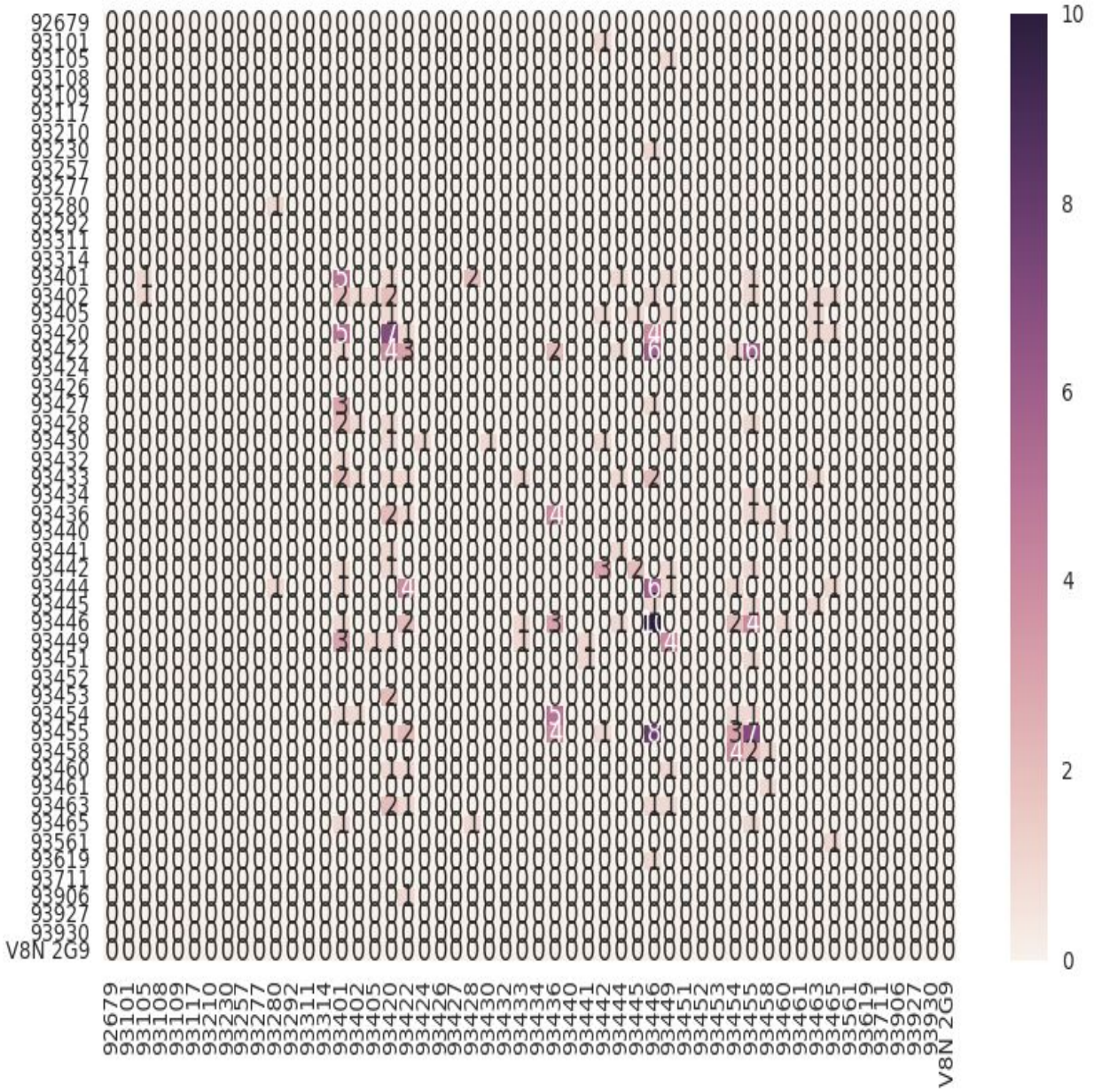
We used K-Nearest Neighbors, SVM, Decision Trees, Random Forest, Multi-layer Perception, Adaboost, and Gaussian classification methods from Sci-kit Learn for our analysis. Since our data are not very linearly separable, we have the intuition that Random Forest will most likely dominate in terms of accuracy. Our process of trying classifiers was automated and selected the best performing classifier in terms of accuracy. The system took in the predictors and the response columns and ran each classifier over the powerset of the predictors. This gave us an understanding of how each classifier performed with different subsets of the data. We also enabled the bootstrap feature while training the classifiers to compensate for our small data set. After performing our batch testing, we confirmed our intuition was correct and that random forest performed the best.

Another approach we took was taking the highest performing classifiers and use a voting classifier. Ultimately we found that the voting classifier underperformed the classifiers on their own.

Data Analysis

Predicting Zip Code

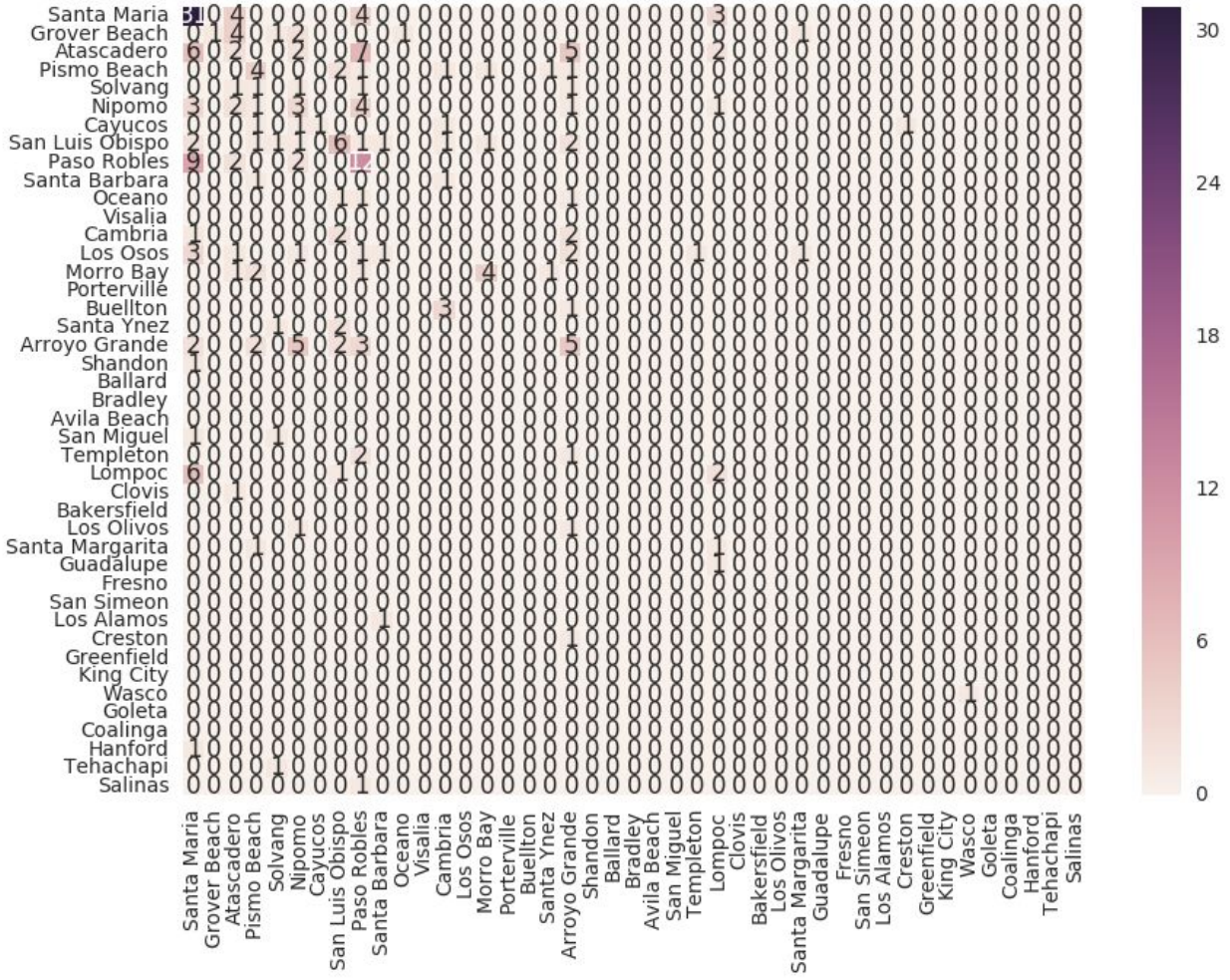
With our classifying system setup we decided to investigate how well we were able to predict zip code, city and housing price class. When predicting zip code we were able to achieve 21.5% accuracy in our model using BR, Footage, Price, PricePerSqft with the random forest classifier.



Since there are too many classes and too little data points for the classifier to train on, the result was expectedly bad.

Predicting City

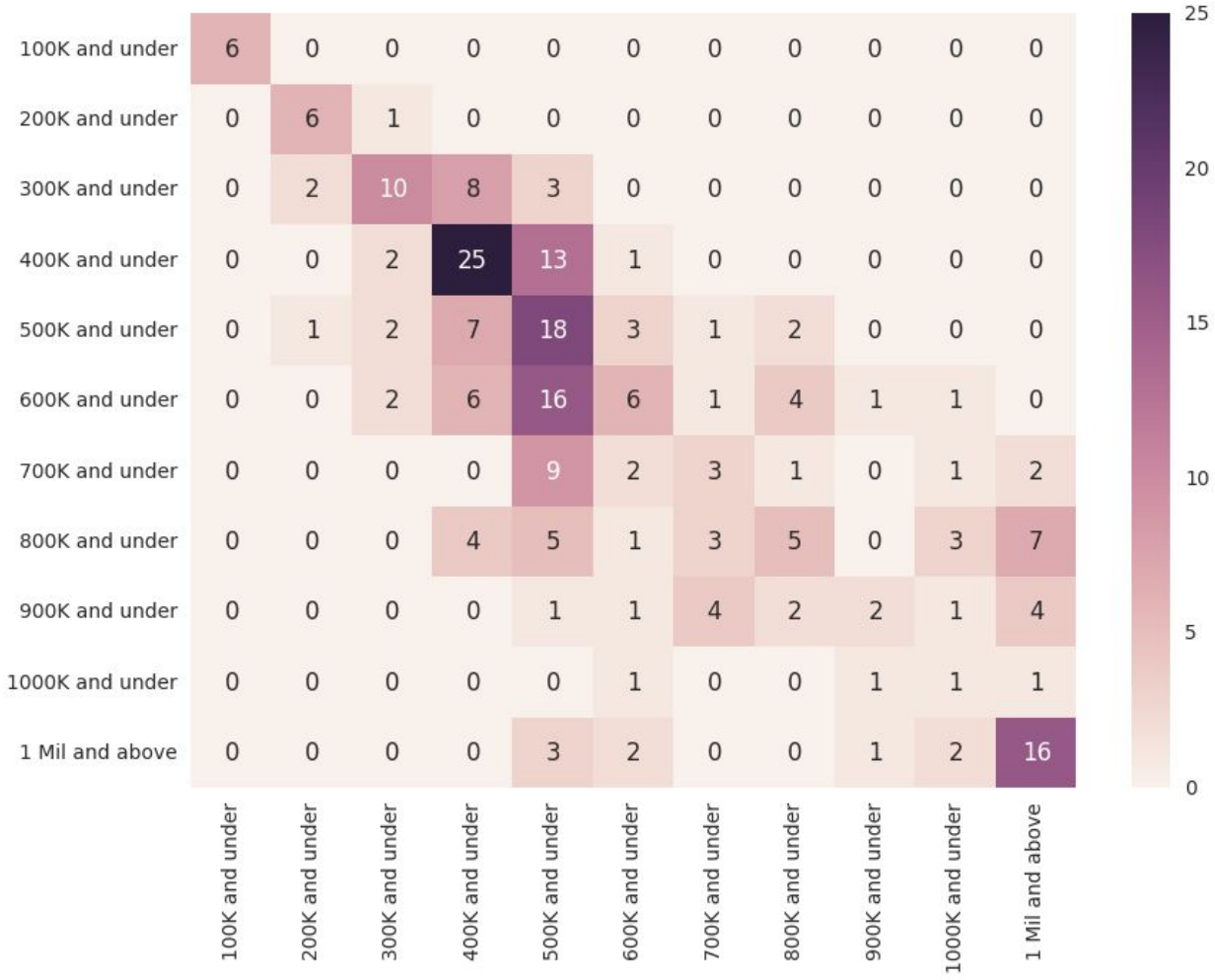
In terms of predicting the city, we were a little bit more accurate with 30.3% accuracy using Random Forest with Price, BR, and Footage as our features. We account this increase in accuracy comparing to zip code due to the fact that we have less classes to classify the data into.



The city prediction heat map shows us that we are predicting more houses in Santa Maria than in other cities. This is because Santa Maria have 217 entries or about 20% of the data, so it is not to our surprise that we are able to predict more houses to be in Santa Maria than others. But still, we are only accurately pointing out 14% of the houses in Santa Maria.

Predicting Price Category

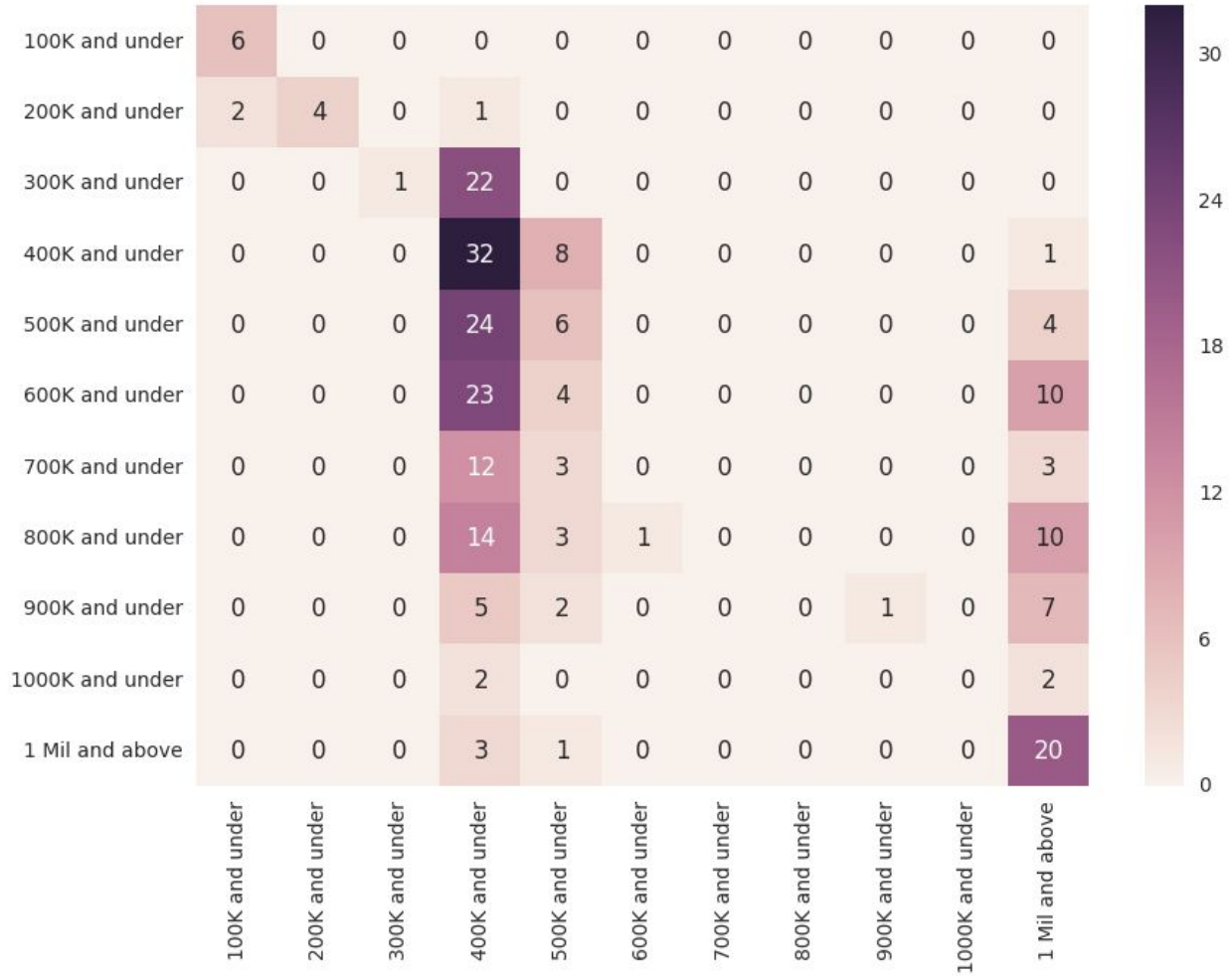
Finally, we had the best results with predicting price class. We split up the price into 11 classes, the base class is $0 - $100K, and every class differs by $100K. The last class (class 11) is for houses costing more than $1 million. We were able to achieve about 41% accuracy with Random Forest classifier with BR, Bath, Price Per Square Foot, and City as features.



In the above heat map, we can see that our classifier is doing fairly well for houses in both ends of the price range, and it runs into problems in the middle (from $300K to $600K). We've noticed that a lot of the data in those ranges are fairly similar in terms of the Bedroom and Bathroom count.

Voting for Price Range

We tried to use voting to see if we can get better results for categorizing price range. However, to our surprise, the accuracy decreased to about 29% comparing to pure Random Forest.



We observed that voting does not increase the accuracy of our model for all the variables we were interested in classifying.

Conclusion

After experimenting with the classifiers to try and answer various questions we found that the automation approach allowed us to quickly iterate and explore new questions as they came up. Using the power set on the predictors allowed us to explore all possible combinations of features to use to provide the best classification result.

We also found that voting can yield worse results when used with inaccurate classifiers. Since our classifiers were not performing well to begin with, they probably not only didn’t reach majority consensus, but voted very sparsely.

Even though the accuracy for each of these results seems low, it's understandable given the nature of the features we had access to. Bedrooms and Bathroom count are very similar across all of our data points, and the other quantitative features we had were at least a factor of 100 bigger than the rest. We should have done preprocessing on the data to normalize all the features to improve our accuracy.