Machine Learning Final Project Report

Team Name: The Neural Netcases
Team Members: Naomi Hachen, Anwesha Das, Deepti Panuganti

PCA

At the very start, we recognized that there were too many features, and that processing any kind of regression or classification on these features would take more processing power than our computers could handle. We therefore started out by performing dimensionality reduction on the data using Principle Component Analysis. Matlab provides a PCA function; however, this function could not handle the data so we constructed our own using Single Value Decomposition, as this is the fastest and most efficient method for computing principle components. To compute the svd, we used fsvd.m, an implementation of randomized SVD as outlined by "Halko, N., Martinsson, P. G., Shkolnisky, Y., & Tygert, M. (2010). (This implementation was shared on stackexchange by “petrichor” on Jan 16, 2011.) Using fsvd, we were able to successfully include as many as 3000 features.

Regression

We next took this dimension-reduced data and applied a series of classification tools. We started off with a simple generalized linear fit, glmfit, and based on our intuition about pricing trends chose to regress the PCA-ed data over a log-normal fit. We used this to pass the first baseline. To add penalty and better fit the data, we switched to performing a linear fit with Lasso penalty. Before deciding on Lasso we tried a variety of regressions and regularizations from the liblinear package, but settled on the lasso fit due to superior performance in cross-validation. The package we used was glmnet, developed by Jerome Friedman, Trevor Hastie, Noah Simon, and Rob Tibshirani and can be found online. To optimize this function we used 2500 principle components.

This algorithm, however, did not capture the differences between the cities and did not account for key variations between groups. We tried to use kmeans to better group the data, but found that clustering the data into sufficient clusters was too slow to work with and did not overall produce strong results in cross-validation. Instead, we decided to group the data into their cities and to run separate PCAs and linear fits for each city. Because the 1st and 4th cities were so small, we grouped them in with the cities which had the most similar mean and standard deviation. For the divided cities, we also were able to far decrease the number of principle components used. We ended up using 250 PCs.

The RMSEs on the predicted data for the main algorithms described above were:
1. Lasso with glmnet on PCAed data divided by city: 0.7707
2. Lasso with glmnet on non-PCAed data divided by city (to show need for PCA): 0.7983
3. Lasso with glmnet on PCAed data, with kmeans clustering: 2.4792

SVM

Using support vector machines significantly improved the prediction compared to the other methods. It was still not enough to beat the baseline but it came very close (RMSE of 0.7498 as compared to 0.7400 which was what we had to beat).

The SVM methods used were the svmtrain and svmpredict from the libsvm library developed by Chih-Chung Chang and Chih-Jen Lin. SVMs generally work better on small data sets compared to the other methods, because it takes into account only parts of the data instead of all of it. So the small number of observations doesn't effect the SVM predictions. So, no grouping of cities had to be done. SVM could be implemented on each city entirely separately.

The next step was to improve the current SVM method. The first thing we tried was using a different kernel. We tried using a linear kernel instead of the RBF kernel. This didn't do as well. In fact it was significantly worse giving an RMSEs of around 0.8 on cross validated data. Usually non-linear kernels work better when the number of observations is larger than the number of features. Since this is the case with our data that would explain why the linear kernel didn't do as well.

So keeping the RBF kernel we instead changed the parameter C (cost). Increasing the value of C causes the SVM to compromise a little on the “maximum” separating hyperplane. It selects smaller margin hyperplanes but avoids misclassifying some points that otherwise would have been given the wrong label. We increased the cost and tested on different values of C between 1 to 60 using cross validation.

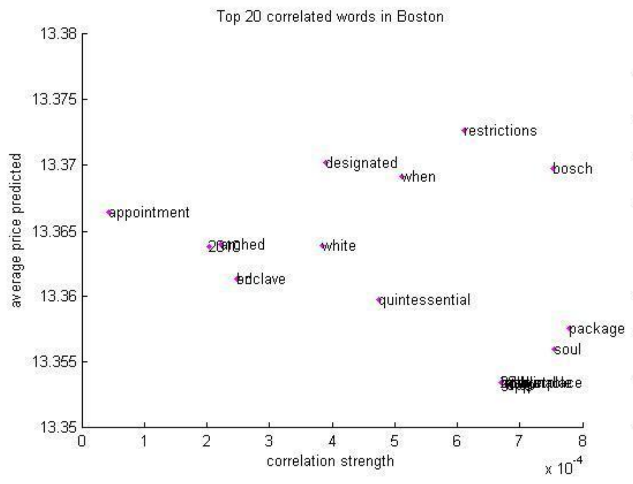
The results obtained by cross validation were most accurate and consistent between 15 to 20. We got the best result with 15 however, so we set the cost to 15, although such a minor change (15-20) in the cost would not have affected the predictions too much. This was the method that passed baseline 2 with an RMSE of 0.7124 (Although, the predictions are slightly different for different runs due to the random component of fsvd, it isn't a very significant difference).

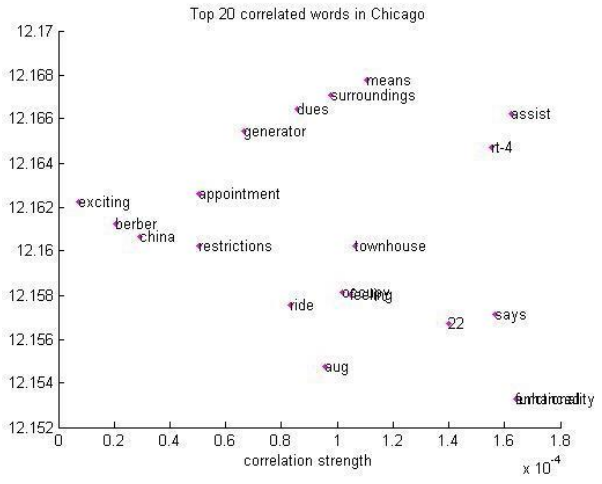
The RMSE's we got on the test data with different costs were approximately as follows:

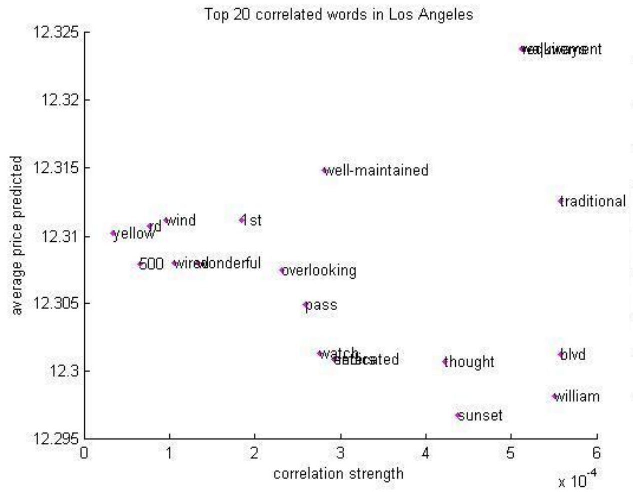
Cost of 15: 0.7103
Cost of 20: 0.7127
Cost of 30: 0.7140
Cost of 60: 0.7233

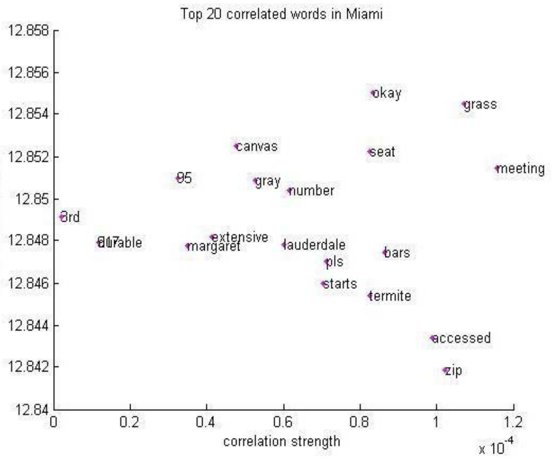
Visualizations

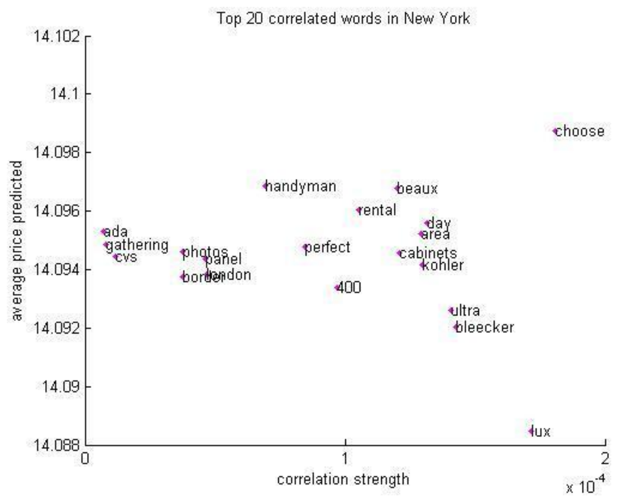
Below please find the top 20 correlated words for each city, and the average price predicted when each word was present. You can see that, though some words like “appointment” and “restrictions” recur, the correlated words definitely vary across city!

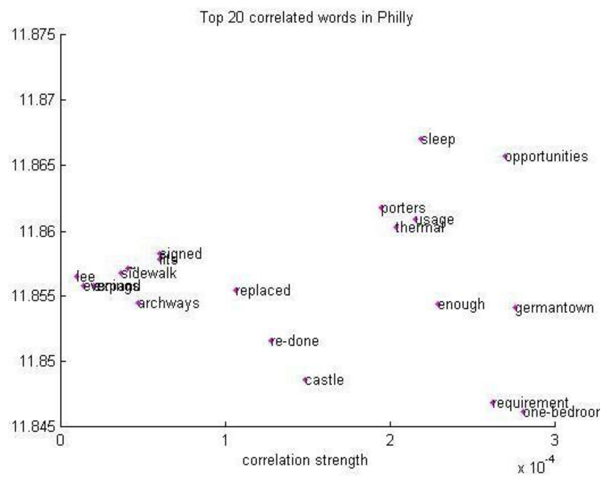


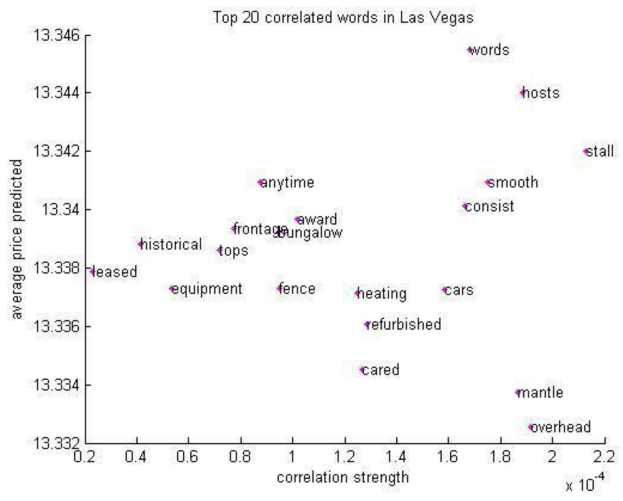







Top 20 correlated words in New York
average price predicted
correlation strength
x 10^-4
14.102
14.1
14.098
14.096
14.094
14.092
14.09
14.088
0
1
2
choose
handyman
beaux
rental
day
area
ada
gathering
cvs
photos
panel
perfect
cabinets
kohler
border
london
400
ultra
bleecker
lux


Top 20 correlated words in Philly
correlation strength
x 10^-4
11.875
11.87
11.865
11.86
11.855
11.85
11.845
0
1
2
3
sleep
opportunities
porters
usage
thermal
signed
fits
sidewalk
fee
evenings
replaced
archways
enough
germantown
re-done
castle
requirement
one-bedroor


Top 20 correlated words in Las Vegas
average price predicted
correlation strength
x 10^-4
13.346
13.344
13.342
13.34
13.338
13.336
13.334
13.332
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
2.2
words
hosts
stall
anytime
smooth
consist
award
frontage
bungalow
historical
tops
leased
equipment
fence
heating
cars
refurbished
cared
mantle
overhead