Predicting Closing Prices on Opendoor Housing Data in McKinney, TX

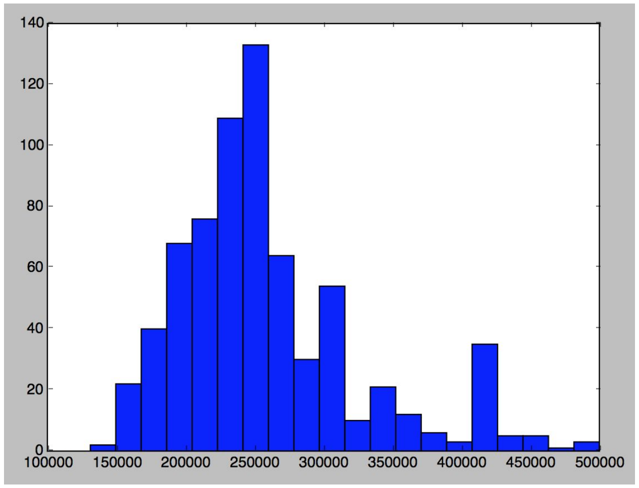
By: Nadin El-Yabroudi and Paul Harrison

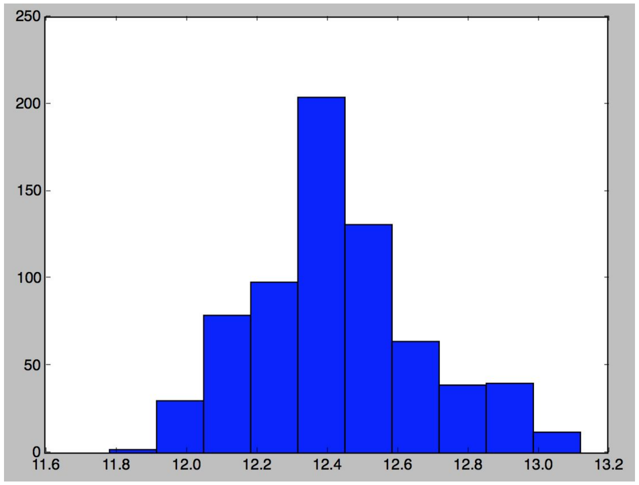
Introduction

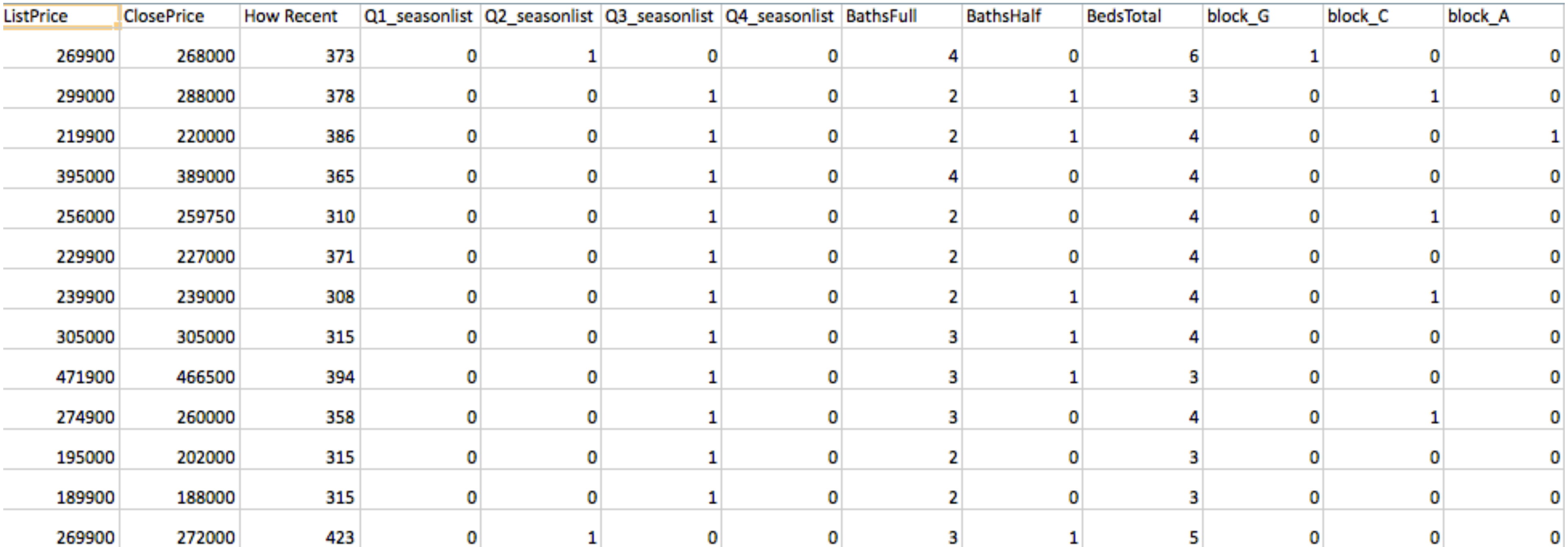
- Goal was to develop a model for homeowners to price their houses once placed on the market
- Giving homeowners a ‘second opinion’ of house’ s market price by learning on close price
- Using Opendoor’s open-sourced real estate data in McKinney,TX mostly applicable to that specific town (due to town specific features e.g. block location)
- Given training data spanning two years (late ‘14 to mid ‘15) our objective was still to model and predict on **most recent** houses
- Training data was fairly small (m≅1100) yet still dimensionality not that high even after preliminary feature processing
(order of ~10 : 1 observations to features)

Data

- Cleaned data of irrelevant features to our prediction scenario before placing on market i.e. no availability of list price, sold date and seller contributions
- Converted categorical features from their list form to binary ones e.g. type of flooring, block, market listing date by quarter
- For classification, bucketed and logged target variable close prices into $40,000 buckets (based off of market research)
- For classification, we also bucketed house's size, living area and age by analysing histograms to fairly represent all price ranges







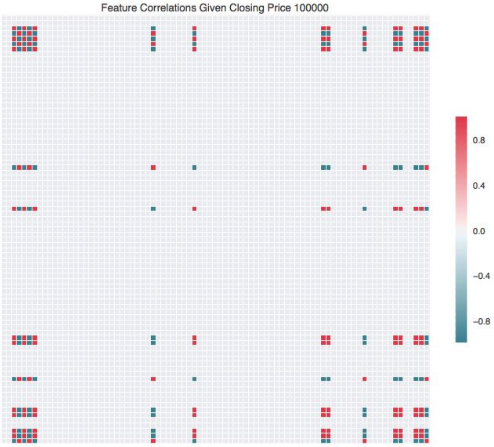
ListPrice	ClosePrice	How Recent	Q1_seasonlist	Q2_seasonlist	Q3_seasonlist	Q4_seasonlist	BathsFull	BathsHalf	BedsTotal	block_G	block_C	block_A
269900	268000	373	0	1	0	0	4	0	6	1	0	0
299000	288000	378	0	0	1	0	2	1	3	0	1	0
219900	220000	386	0	0	1	0	2	1	4	0	0	1
395000	389000	365	0	0	1	0	4	0	4	0	0	0
256000	259750	310	0	0	1	0	2	0	4	0	1	0
229900	227000	371	0	0	1	0	2	0	4	0	0	0
239900	239000	308	0	0	1	0	2	1	4	0	1	0
305000	305000	315	0	0	1	0	3	1	4	0	0	0
471900	466500	394	0	0	1	0	3	1	3	0	0	0
274900	260000	358	0	0	1	0	3	0	4	0	1	0
195000	202000	315	0	0	1	0	2	0	3	0	0	0
189900	188000	315	0	0	1	0	2	0	3	0	0	0
269900	272000	423	0	1	0	0	3	1	5	0	0	0

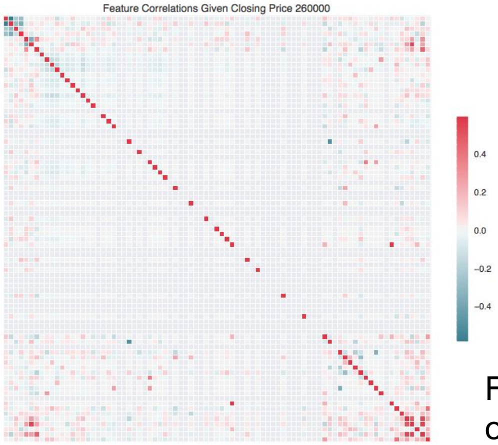
Training

- Split data 70/30 into training and testing sets
- Used cross validation as a proxy for test error on all training models
- Set aside test set to use only at the end once the best models were chosen

Classification

- Baseline model Naive Bayes
- Required discretizing closing price (target values)
- Best classification for Naive Bayes yielded 28% accuracy.
- Decided to try multinomial logistic regression (logit) because of model's flexibility
- Best classification for Logit yielded 48% accuracy, proving that model is more flexible than NB


Feature Correlations Given Closing Price 100000


Feature Correlations Given Closing Price 260000

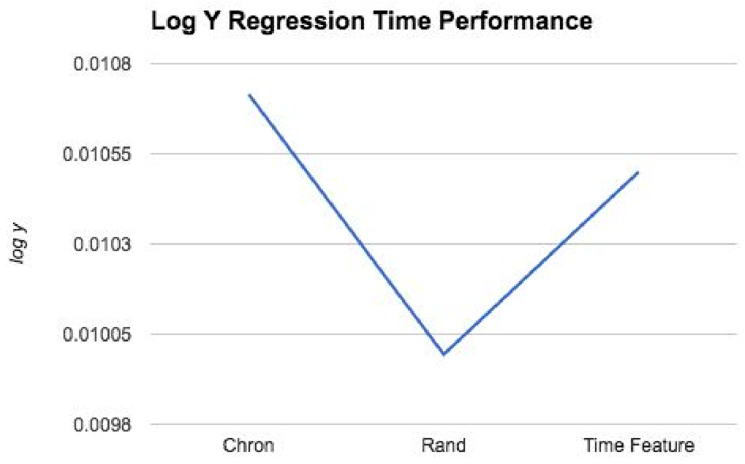
Regression

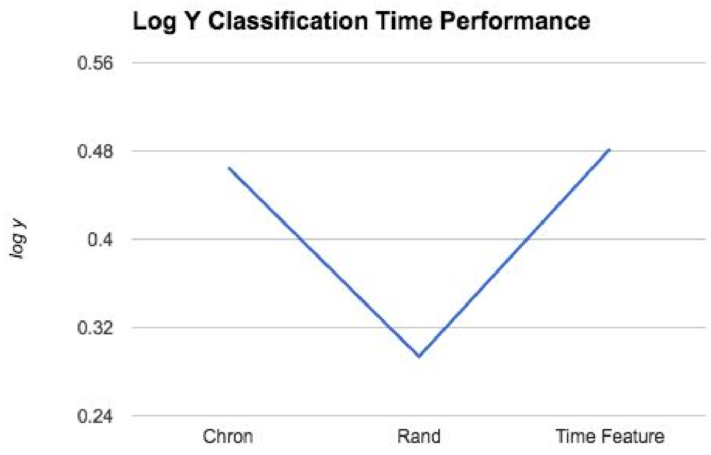
- Decided to use regression to avoid losing information in bucketing.
- Linear regression without regularization produced prediction values extremely high (about 1000 times the price) for some samples.
- Regression permitted for predicting with more granularity the closing price, at best within 0.7%. To compare classification and regression we tried buckets with half the size on classification to see if we could achieve the same granularity. Halfing the buckets significantly lowered accuracy scores for classification.
- Used Mean Square Percentage Error to evaluate regression performance to be consistent with model risk function
- Concluded that Regression is a better model for this data.

Feature correlation plots show high dependency of the features conditional on a closing price

Time Performance

- Compared performance of chronological training set to randomly-timed training set. Time seemed to be important in our results in both regression and classification.
- To account for time we added a time feature to our dataset indicating the amount of time since the house was placed on the market. This could account for market differences in time.
- Results below show that for both regression and classification, the time features helped our models make better predictions. In classification randomized data performed worse than chronological data, but the time feature data set performed better than both. In regression, randomized data performed better than chronological data and the time feature helped find a midpoint between these.





Interaction Terms

- Added interaction terms between features to second degree to explore how the dependence of features could help our model with prediction.
- The addition of an important amount of terms meant that a more aggressive regularization was necessary to avoid overfitting.
- For both regression and classification we see that interactions terms do better than the dataset without interaction terms.
- This confirms that the dependence between variables is important to our modelling.

	Classification	Regression
Interaction Terms	0.4922873809	0.007814019143
No Interaction Terms	0.4802815793	0.01050023212

Regularization

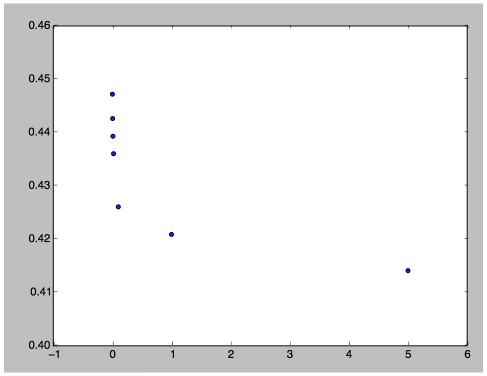
Classification

- Regularizing with L1, we varied the strength of the regularization using different c-values, inversely proportional to regularization.
- Interaction terms show exponential increase in accuracy as regularization was increased to avoid overfitting.
- Non-interaction terms seemed to do best with moderate regularization (c=0.7)
- Interaction terms required higher levels of regularization than non-interaction terms.

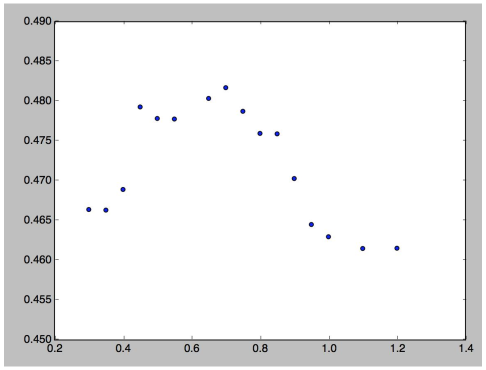
Regression

- Regularizing with lasso, we varied the strength of regularization using different alpha-values.
- Interaction terms show how adding small amounts of regularization increase model's performance tremendously
- Again see similar result without interactions yet far less regularization needed i.e. 0.0001 instead of 0.4 due to underfitting
- For both models lasso (L1) regularization performs far better than ridge (L2) due to more aggressive L1 penalty ensuring model does **not** overfit

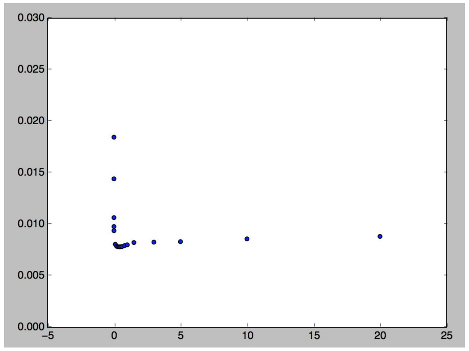
C-values vs. Accuracy for Interaction Terms



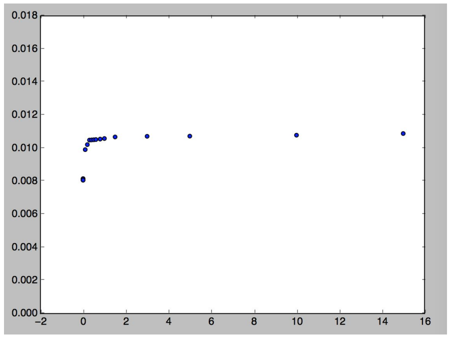
C-values vs. Accuracy for non-Interaction Terms



Alpha-Values vs. Mean Squared Percentage Error for Interaction Terms

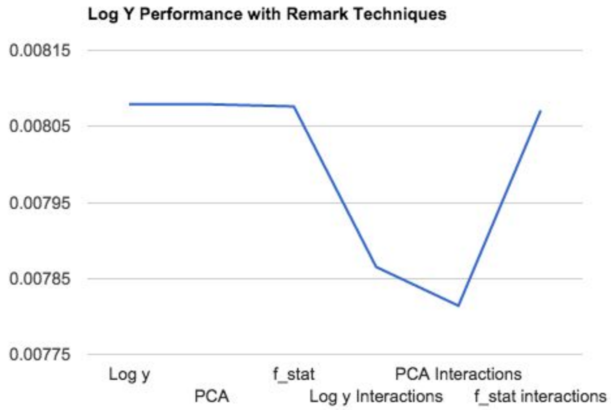


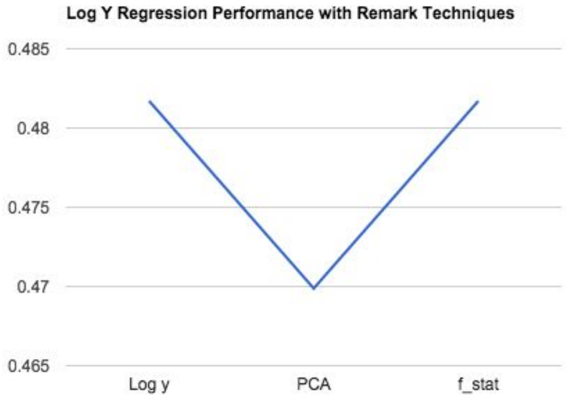
Alpha-values vs. Mean Squared Percentage Error for Non-Interaction Terms



Remarks

- Attempted to do basic NLP on house remarks to improve classification.
- Used bag of words to count frequency of words
- Used TF-IDF to remove words which appeared more that 5% of the time and remove common English words
- Reduced dimensionality of resulting matrix to leave only 10% of features with two methods: PCA and ANOVA F-value (f_stat)
- ANOVA F-value model selected the top 10 percentile of features which give the most information about the closing price
- Appended resulting matrices to other feature matrices to analyze results.
- On the results (on right) similar regularization was used when PCA and f_stat were appended to a model.
- Overall the remarks had minimal effects on our model. PCA performed better than f_stat for regression, but f_stat performed better for classification than PCA.

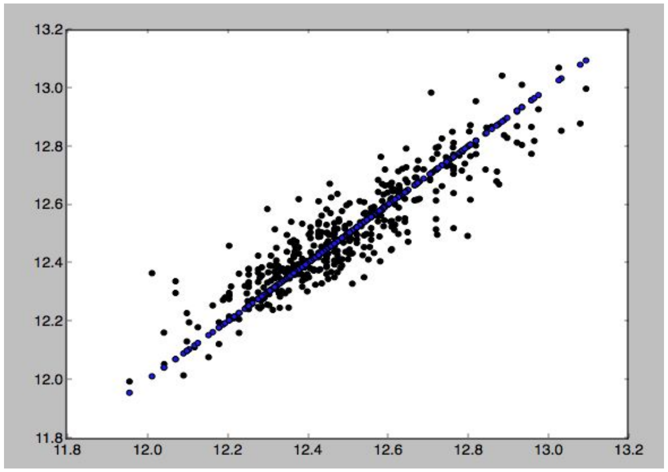




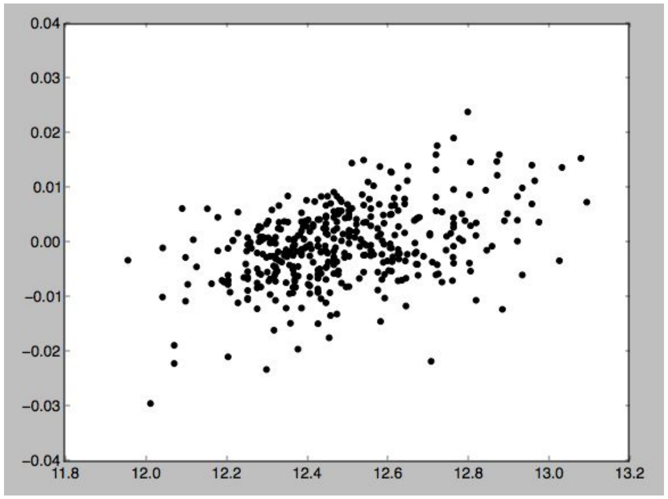
Testing and Conclusion

- Achieved strong model using lasso linear regression with aggressive regularization, including interaction terms, and predicting on the log value of the closing price.
    - ○ Test Error ≅ 0.00709 = 0.709%
    - ○ Confidence ± ≅ $3452
- Residual plot shows model performance does best on middle range of values, and performs worse at extreme values
- To try to improve model attempted linear regression with kernels, and random forests, both of which performed similar to best linear regression model.
- Again found our NLP modelling to have little to no impact on model performance, which remarks are unreliable
- Further research would be to look at ensembling techniques + tranche based training

Predicted (black) vs. Actual Values (blue)



Residual Plot Log Y Interaction Terms


0.04
0.03
0.02
0.01
0.00
−0.01
−0.02
−0.03
−0.04
11.8
12.0
12.2
12.4
12.6
12.8
13.0
13.2