TWO SIGMA CONNECT: RENTAL LISTING INQUIRIES

HOW MUCH INTEREST WILL A NEW RENTAL LISTING ON RENTHOP RECEIVE?

TOPIC

- The objective of this Kaggle Competition is to predict how popular an apartment rental listing is based on the listing content like text description, photos, number of bedrooms, price, etc.
- The data comes from renthop.com, an apartment listing website
- These apartments are located in New York City

PROBLEM STATEMENT

- Supervised learning multi-class classification problem
- Given rental listings that are comprised of 3 types of data:
  - Numeric – number of bedrooms, bathrooms, monthly rent, geographic coordinates
  - Text – apartment features, detailed apartment description
  - Photos – listing photos from apartment
- We wish to determine which aspects of a rental listing best predict the interest level (low, medium, high) that a listing receives

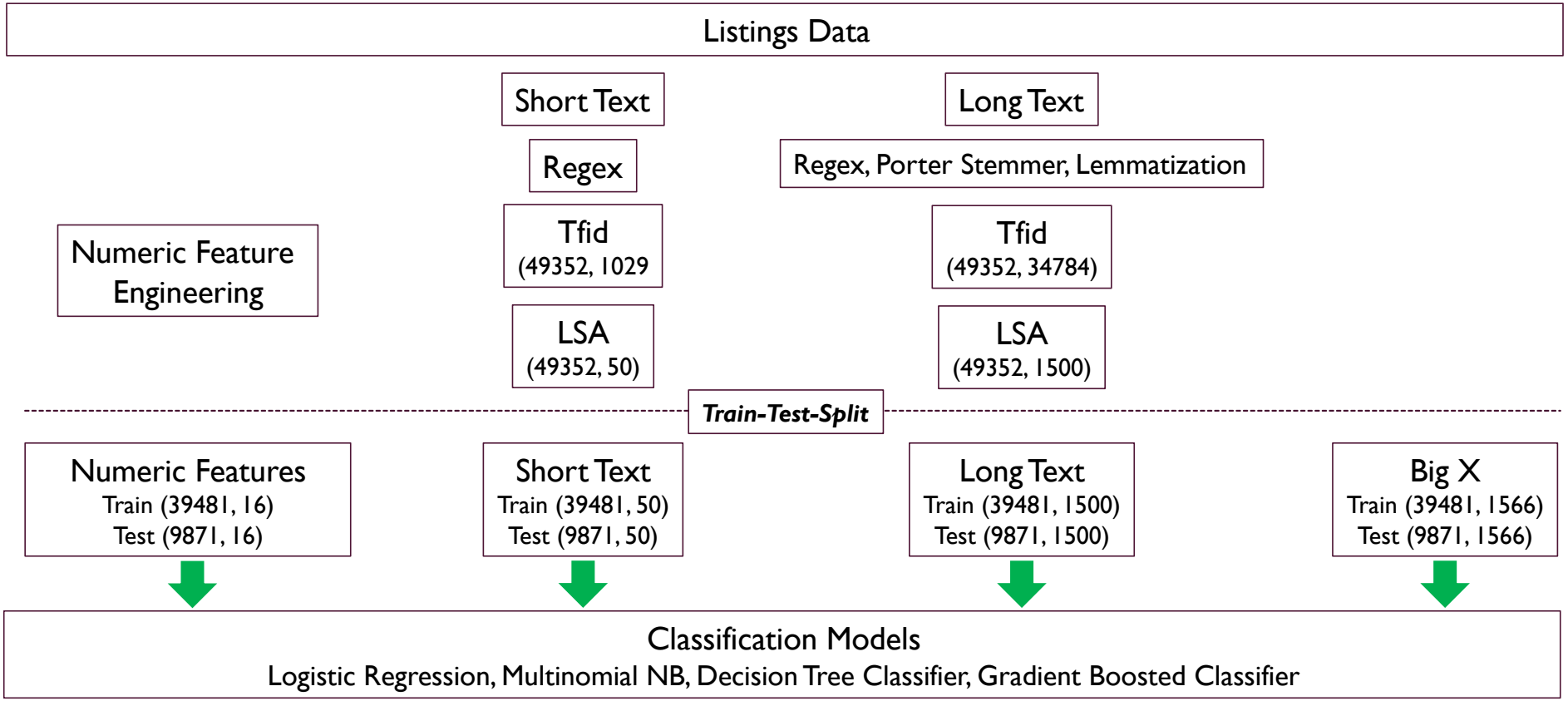
DATA

	10	10000
bathrooms	1.5	1
bedrooms	3	2
building_id	53a5b119ba8f7b61d4e010512e0dfc85	c5c8a357cba207596b04d1afd1e4f130
created	2016-06-24 07:54:24	2016-06-12 12:19:27
description	A Brand New 3 Bedroom 1.5 bath ApartmentEnjoy ...	
display_address	Metropolitan Avenue	Columbus Avenue
features	[]	[Doorman, Elevator, Fitness Center, Cats Allow...
interest_level	medium	low
latitude	41	41
listing_id	7211212	7150865
longitude	-74	-74
manager_id	5ba989232d0489da1b5f2c45f6688adc	7533621a882f71e25173b27e3139d83d
photos	[https://photos.renthop.com/2/7211212_1ed4542e...	[https://photos.renthop.com/2/7150865_be3306c5...
price	3000	5465
street_address	792 Metropolitan Avenue	808 Columbus Avenue

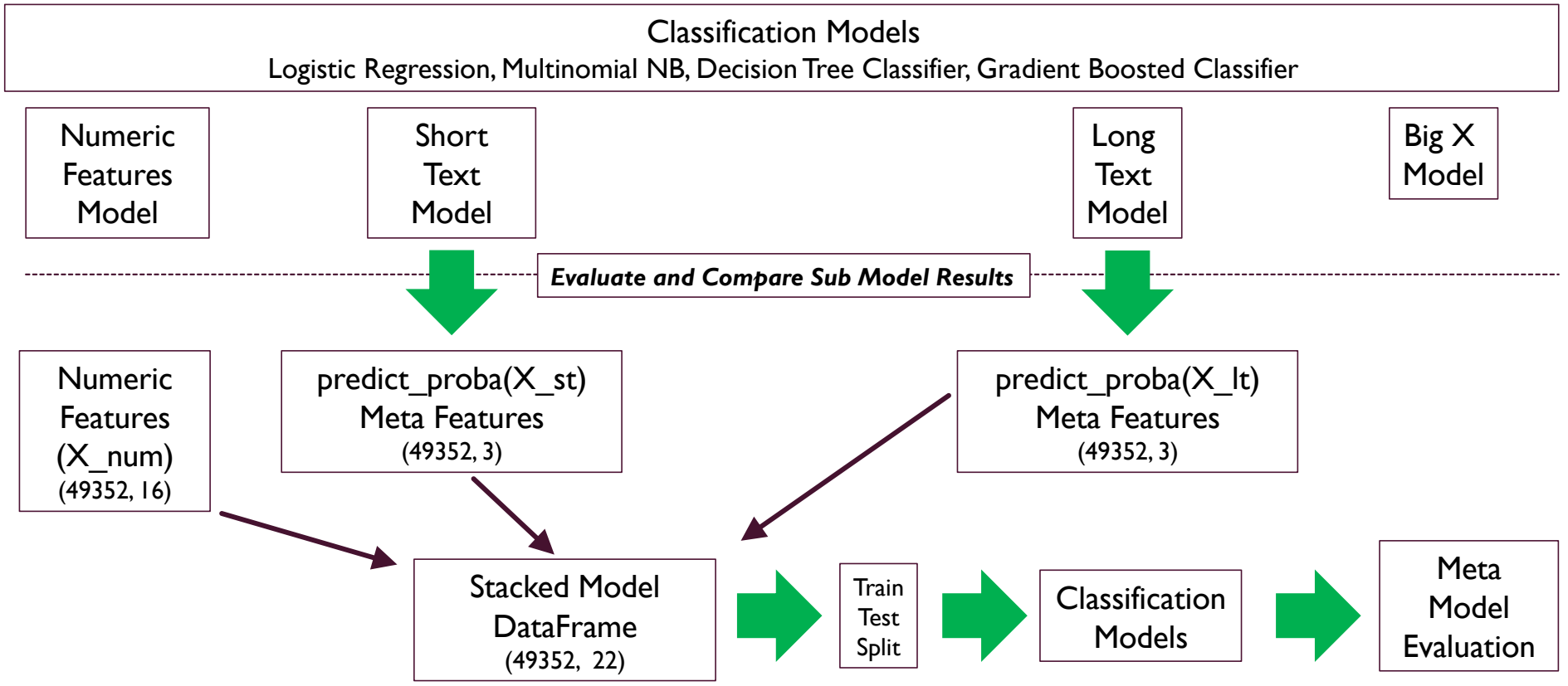
APPROACH

- Apply feature engineering and automated feature selection techniques to each sub-model
- Develop 'sub' models for each data type to determine effectiveness of each data type on its own and compare to 'master' or 'Big X' model where all features included
- Use the predictions of the sub-models features (i.e. meta features) for a stacked (aka 2nd level) model
- In order to assess the validity of our models, we will use Accuracy and Log Loss as our success metrics

MODELLING WORKFLOW I



MODELLING WORKFLOW II

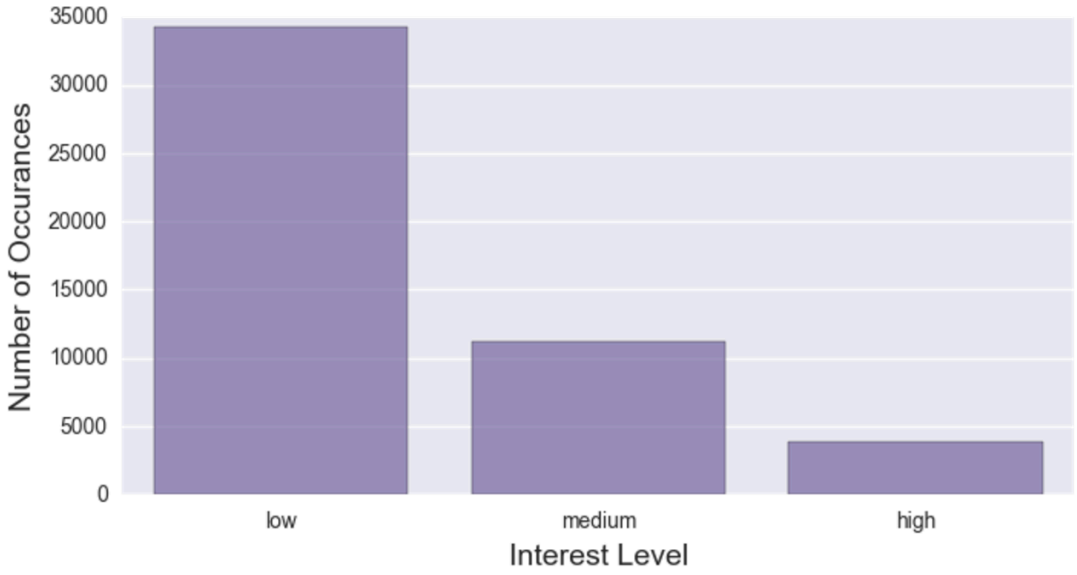


RESULTS | CONCLUSIONS

- Occam's Razor
  - Simplest approach – numeric features – delivered best scores
  - NLP on text data yielded no accuracy improvement
- Overall results fairly disappointing – accuracy not meaningfully above baseline accuracy

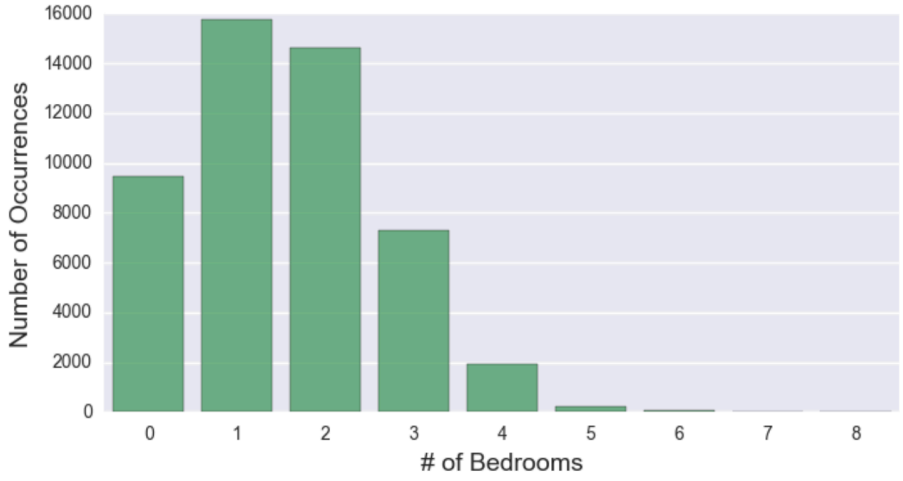
EDA | VISUALIZATIONS

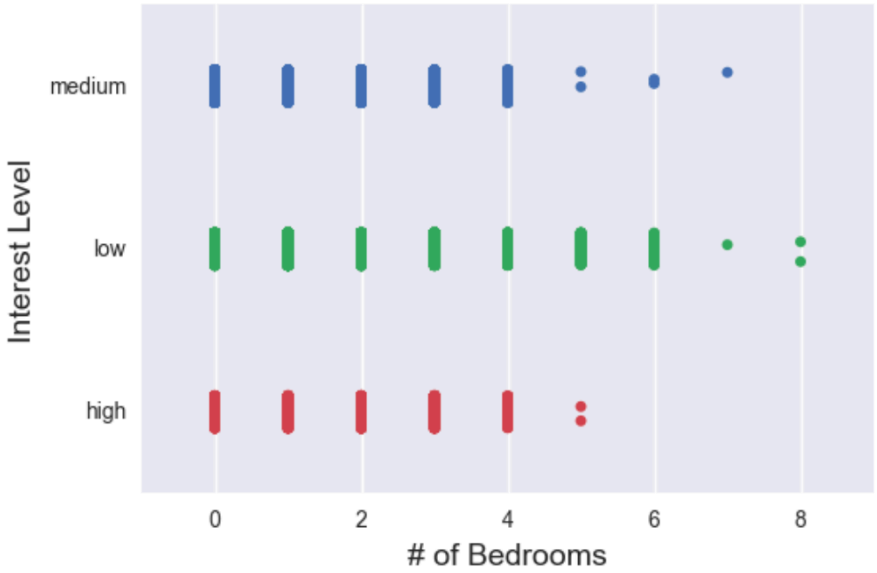
- Imbalanced classes – 'Low' interest level accounted for 70% of the observations, 'Medium' – 23%, 'High' – 7%



EDA | VISUALIZATIONS

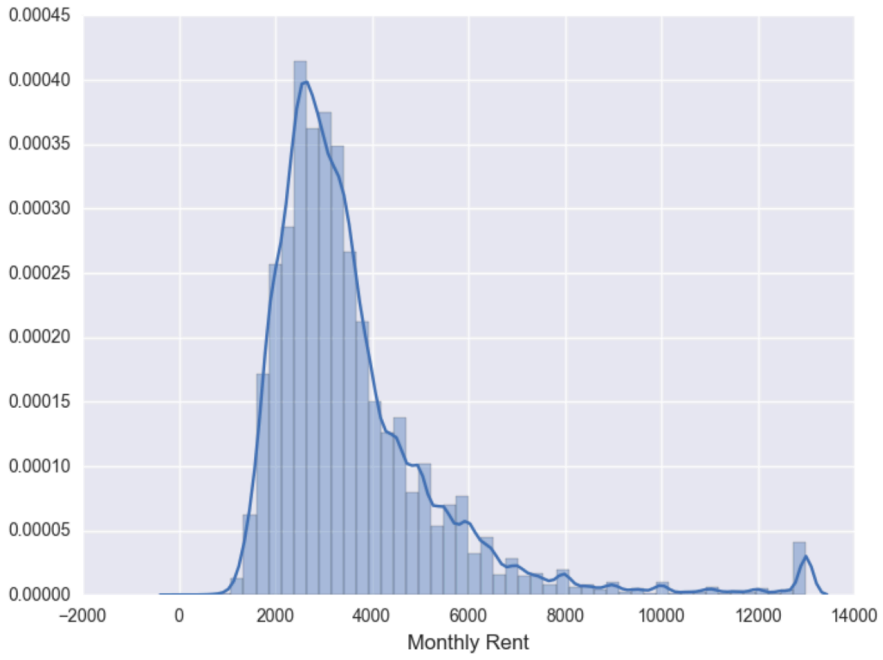
- In general, EDA and visualizations did not yield a lot of insight – interest level is fairly evenly distributed by number of bedrooms – similar results for the other numerical features

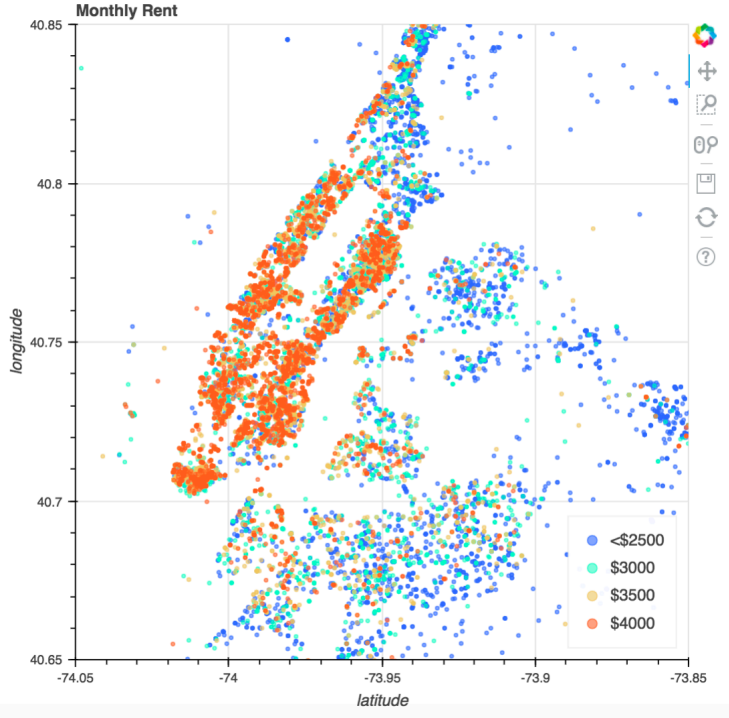




EDA | VISUALIZATIONS

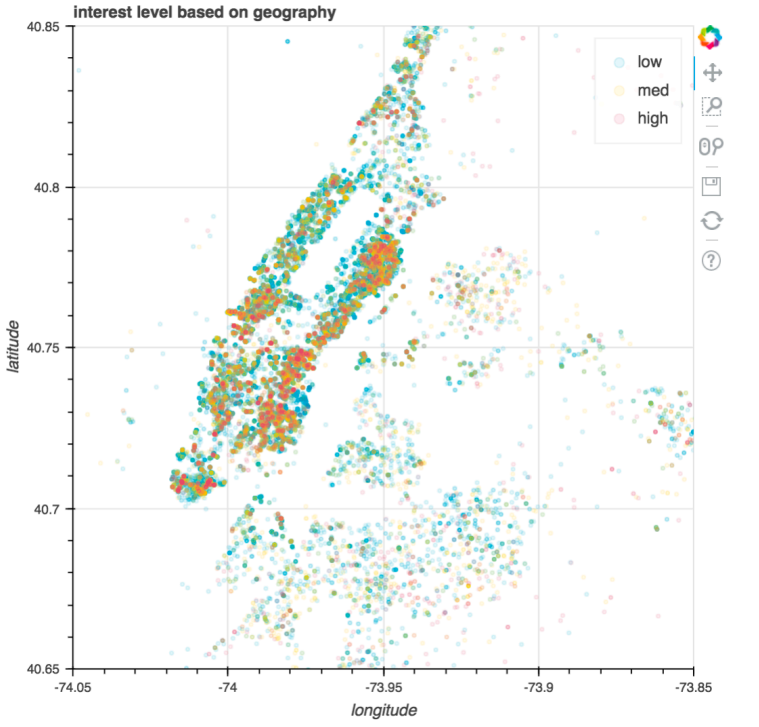
- Monthly rent was an important feature (per our models) – distribution was skewed to the right

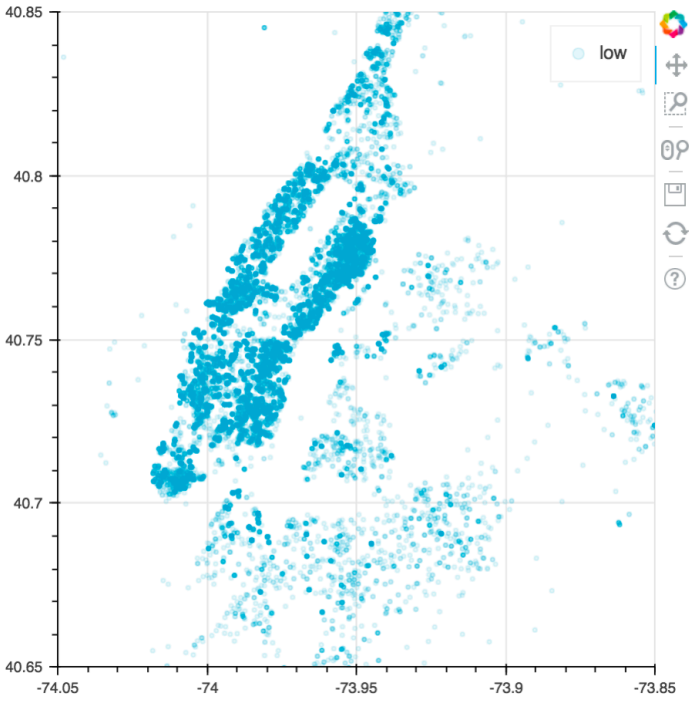




EDA | VISUALIZATIONS

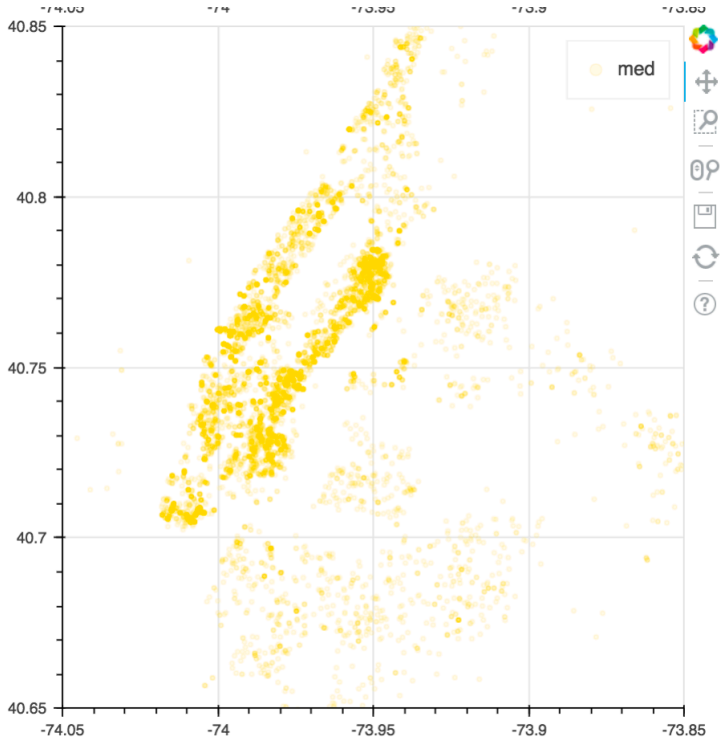
- Location, location, location - meh

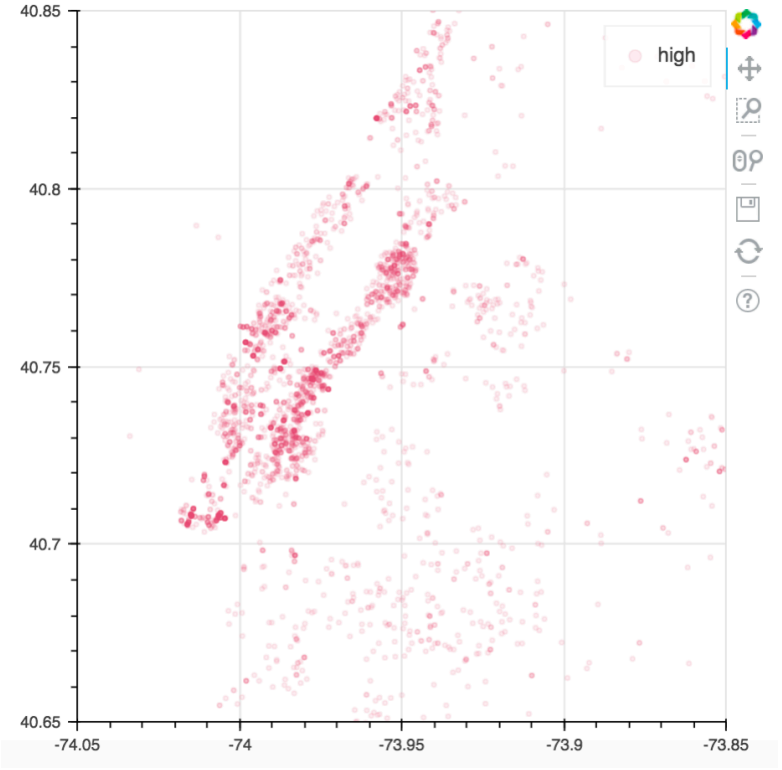




EDA | VISUALIZATIONS

- Location, location, location - meh



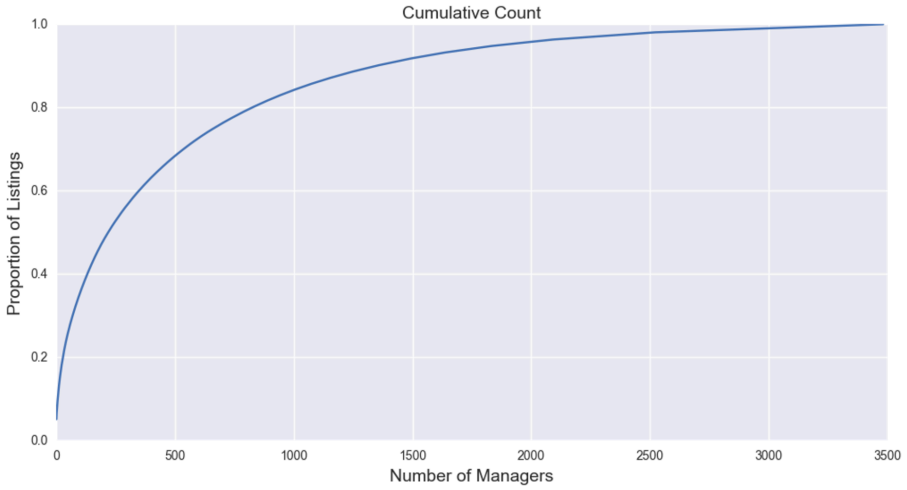


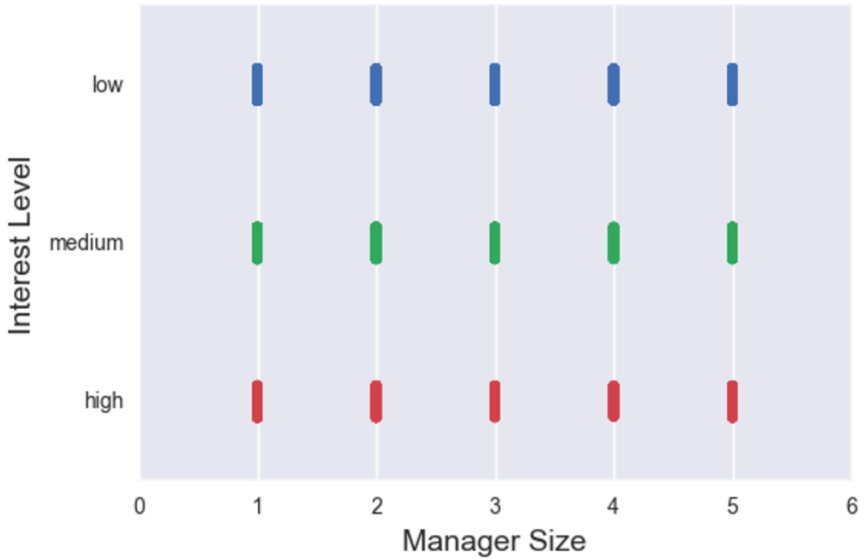
BASIC NUMERIC FEATURE ENGINEERING

- Simple feature engineering on numeric features:
  - Number of photos
  - Number of features listed
  - Length (number of words) of apartment description
  - Binned manager_id by number of listings per manager
  - Converted listing date created into a pandas datetime object and binned the day of the month the listing was created

ENGINEERED FEATURE VISUALIZATIONS

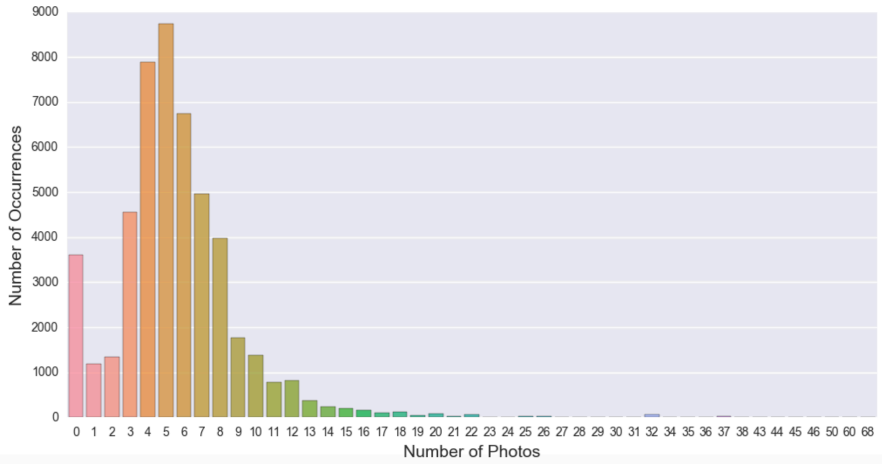
- Not much added insights

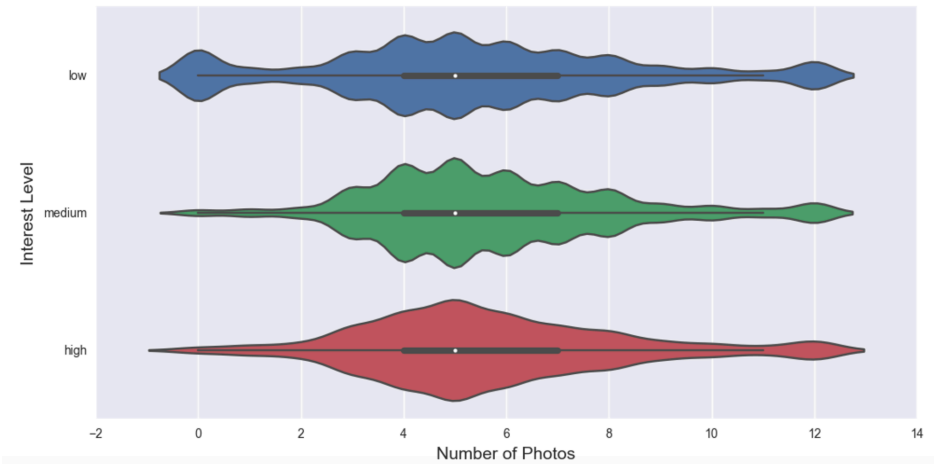




ENGINEERED FEATURE VISUALIZATIONS

- Number of photos in listing



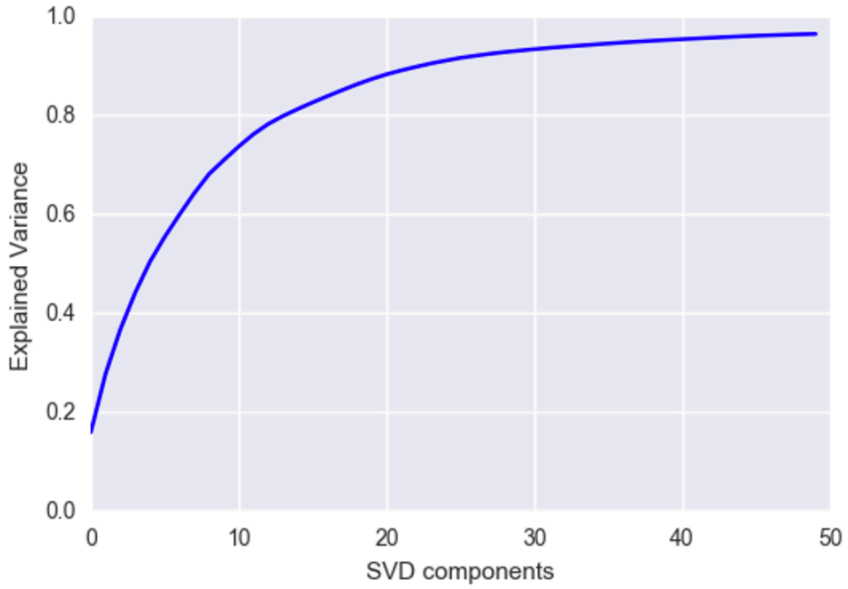


TEXT FEATURES

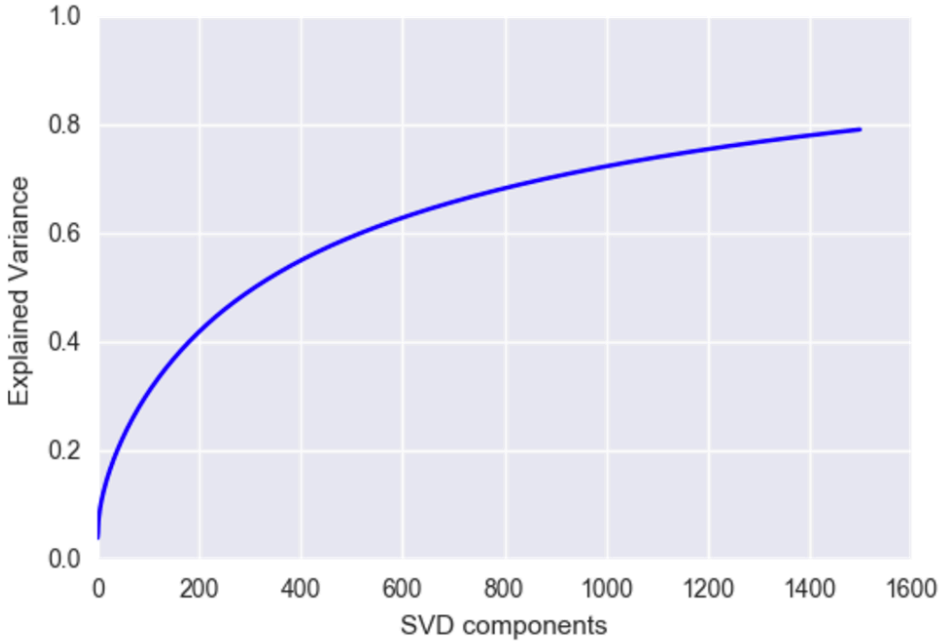
- 'short_text' – list of apartment features (doorman, elevator, etc.)
  - Average of 5 features listed per listing
- 'long_text' – detailed apartment description prepared by broker
  - Average of 90 words per listing, 'raw' corpus ~90K unique words
- Prepared text data by cleaning up http tags, e-mail addresses and other potential noise via regex
- Applied Porter Stemmer and Lemmatization
- TfidfVectorizer
- Automatic feature selection via TruncatedSVD
- Modelled using Logistic Regression, Decision Tree Classifier and Gradient Boosting Classifier

TEXT FEATURE SELECTION

- With Truncated SVD reduced number of short_text features from > 1000 (post Tfid) to 50



- With Truncated SVD reduced number of long_text features from > 32K (post Tfid) to 1500



X_NUM SUB-MODEL RESULTS

- Gradient Boosted Classifier model yielded the best results

	Accuracy		Log Loss	
Model	**Train**	**Test**	**Train**	**Test**
Logistic Regression	0.6851	0.6922	0.7639	0.7688
Decision Tree Classifier	0.7055	0.6976	0.6917	0.7336
Gradient Boosted Classifier	**0.7767**	**0.7193**	**0.5264**	**0.6647**

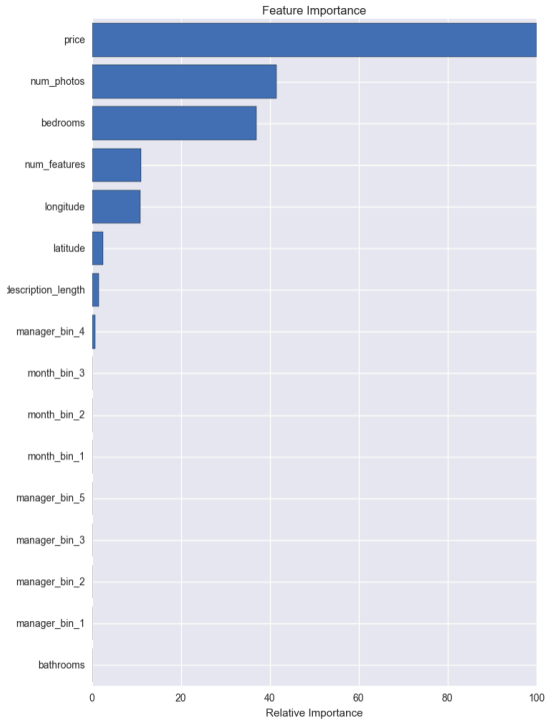
X_NUM SUB-MODEL FEATURE IMPORTANCE

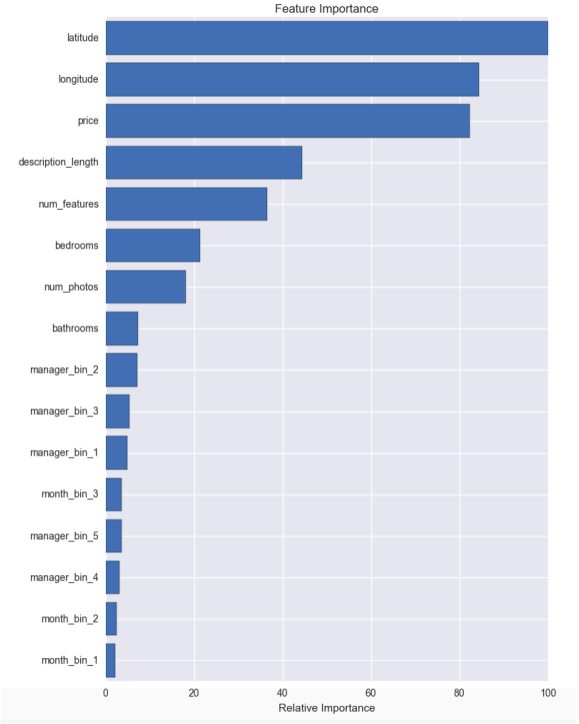
- Logistic Regression with Lasso Feature Coefficients

	Feature	coef_high	coef_low	coef_medium	high_abs_coef	low_abs_coef	med_abs_coef
2	price	-0.123141	0.238975	-0.152796	0.123141	0.238975	0.152796
0	bathrooms	-0.034013	0.050512	-0.041195	0.034013	0.050512	0.041195
7	longitude	0.025883	-0.008005	0.011990	0.025883	0.008005	0.011990
3	num_photos	0.025404	-0.064929	0.050286	0.025404	0.064929	0.050286
12	manager_bin_5	-0.019418	0.043558	-0.028478	0.019418	0.043558	0.028478
4	num_features	-0.015543	-0.051190	0.062809	0.015543	0.051190	0.062809
6	latitude	-0.013946	0.005717	-0.008155	0.013946	0.005717	0.008155
9	manager_bin_2	0.013721	-0.048916	0.037766	0.013721	0.048916	0.037766
15	month_bin_3	-0.013227	0.016471	-0.005910	0.013227	0.016471	0.005910
1	bedrooms	0.013056	-0.087178	0.067962	0.013056	0.087178	0.067962
13	month_bin_1	0.012039	-0.018708	0.008659	0.012039	0.018708	0.008659
5	description_length	0.010005	-0.072147	0.069865	0.010005	0.072147	0.069865
8	manager_bin_1	0.004093	0.029940	-0.033786	0.004093	0.029940	0.033786
10	manager_bin_3	0.000494	-0.013699	0.013629	0.000494	0.013699	0.013629
14	month_bin_2	0.000426	0.002642	-0.003236	0.000426	0.002642	0.003236
11	manager_bin_4	0.000347	-0.010477	0.010382	0.000347	0.010477	0.010382

X_NUM SUB-MODEL FEATURE IMPORTANCE

- Decision Trees. Classifier versus Boosted Gradient Classifier





X_SHORT_TEXT SUB-MODEL RESULTS

- Suspect results or no real information provided by text data

	Accuracy		Log Loss	
Model	**Train**	**Test**	**Train**	**Test**
Logistic Regression	0.6952	0.6923	0.9079	0.9095
Decision Tree Classifier	0.6953	0.6923	0.7857	0.7921
Gradient Boosted Classifier	0.6953	0.6923	0.8096	0.8133
Multinomial Naïve Bayes Classifier	0.6982	0.6862	1.0162	1.0162

X_LONG_TEXT SUB-MODEL RESULTS

- Suspect results or no real information provided by text data

	Accuracy		Log Loss	
Model	**Train**	**Test**	**Train**	**Test**
Logistic Regression	0.6953	0.6923	0.9096	0.9113
Decision Tree Classifier	0.6953	0.6923	0.7874	0.7924
Gradient Boosted Classifier	0.6953	0.6923	0.8092	0.8134
Multinomial Naïve Bayes Classifier	0.6958	0.6902	0.7979	0.8033

BIG 'X' MODEL SUMMARY RESULTS

- Simple beats complex

	Accuracy		Log Loss	
Model	**Train**	**Test**	**Train**	**Test**
Logistic Regression	0.6951	0.6922	0.7639	0.7688
Decision Tree Classifier	0.6998	0.6924	0.7298	0.7436
Gradient Boosted Classifier	0.6953	0.6923	0.7790	0.7846

STACKED MODEL RESULTS SUMMARY

- Used 'y_pred_proba' from text models as additional features to base 'X_num' model
- Simplest 'X_num' model performs the best
- Text features add no value

	Accuracy		Log Loss	
Model	**Train**	**Test**	**Train**	**Test**
Logistic Regression	0.6957	0.6900	0.7634	0.7705
Decision Tree Classifier	0.7029	0.6954	0.6879	0.7034
Gradient Boosted Classifier	0.7798	0.7099	0.5175	0.6707

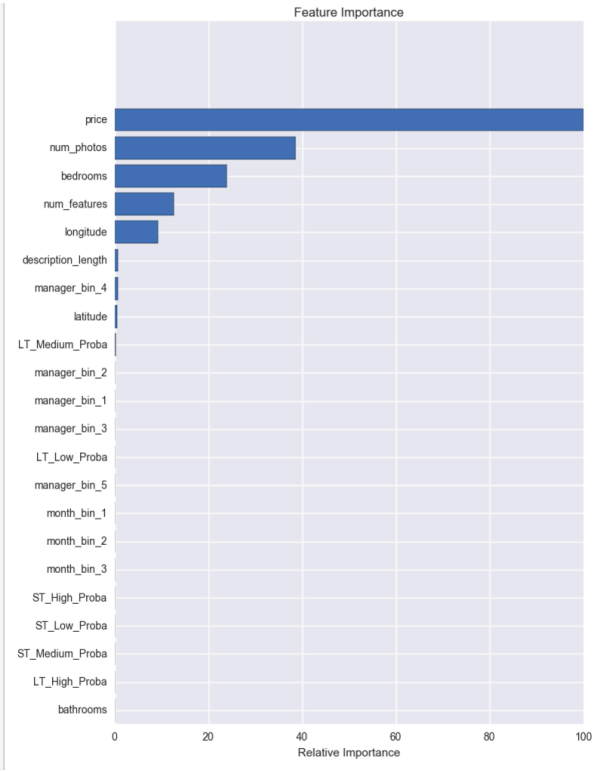
STACKED MODEL FEATURE IMPORTANCE

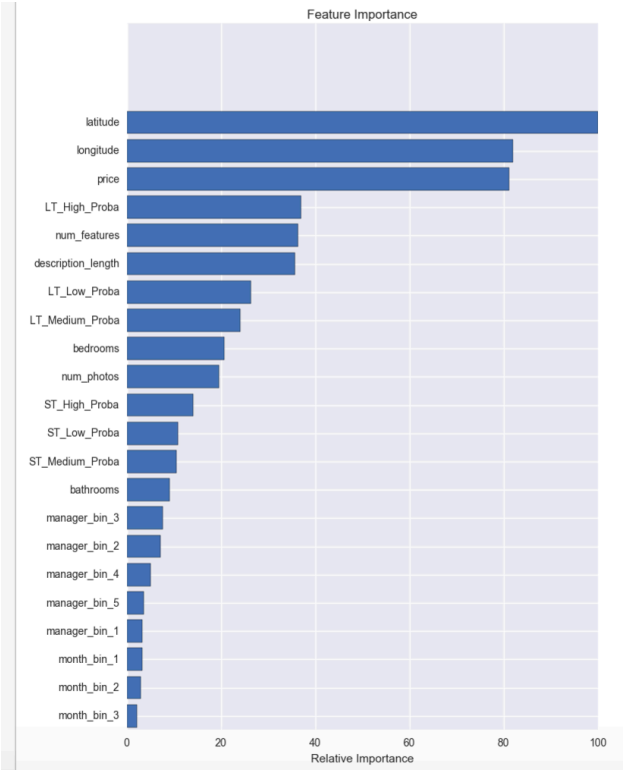
- Logistic Regression with Lasso

	Feature	coef_high	coef_low	coef_medium	high_abs_coef	low_abs_coef	med_abs_coef
2	price	-0.121728	0.240330	-0.154632	0.121728	0.240330	0.154632
0	bathrooms	-0.033000	0.047744	-0.038338	0.033000	0.047744	0.038338
3	num_photos	0.026508	-0.067933	0.052913	0.026508	0.067933	0.052913
7	longitude	0.025782	-0.008226	0.012276	0.025782	0.008226	0.012276
12	manager_bin_5	-0.019055	0.042070	-0.027190	0.019055	0.042070	0.027190
1	bedrooms	0.017996	-0.090431	0.067944	0.017996	0.090431	0.067944
9	manager_bin_2	0.015569	-0.049402	0.036571	0.015569	0.049402	0.036571
6	latitude	-0.014013	0.005369	-0.007739	0.014013	0.005369	0.007739
5	description_length	0.012549	-0.073800	0.069926	0.012549	0.073800	0.069926
15	month_bin_3	-0.012381	0.017130	-0.007343	0.012381	0.017130	0.007343
4	num_features	-0.012171	-0.052933	0.062589	0.012171	0.052933	0.062589
13	month_bin_1	0.009615	-0.020488	0.012762	0.009615	0.020488	0.012762
8	manager_bin_1	0.003147	0.030913	-0.033968	0.003147	0.030913	0.033968
14	month_bin_2	0.002104	0.003662	-0.005807	0.002104	0.003662	0.005807
10	manager_bin_3	-0.000684	-0.013106	0.014145	0.000684	0.013106	0.014145
20	LT_Low_Proba	-0.000414	0.000194	-0.000243	0.000414	0.000194	0.000243
17	ST_Low_Proba	-0.000393	0.000163	-0.000232	0.000393	0.000163	0.000232
11	manager_bin_4	0.000361	-0.010171	0.010054	0.000361	0.010171	0.010054
18	ST_Medium_Proba	-0.000170	0.000104	-0.000112	0.000170	0.000104	0.000112
21	LT_Medium_Proba	-0.000158	0.000050	-0.000069	0.000158	0.000050	0.000069
16	ST_High_Proba	-0.000098	0.000037	-0.000045	0.000098	0.000037	0.000045
19	LT_High_Proba	-0.000090	0.000060	-0.000076	0.000090	0.000060	0.000076

STACKED MODEL FEATURE IMPORTANCE

- Decision Trees versus Boosted Gradient Classifier





SUGGESTIONS FOR FURTHER RESEARCH

- Reverse geocoding on geographic coordinates – extract neighborhoods and distance from center of respective neighborhoods from coordinates
- Machine learning on listing photos
- Beating the dead horse that is text data – tune n_gram parameter in TfidfVectorizer to capture potentially useful features like square footage information, other useful 1-3 word combinations (computationally expensive), extracting broker names, additional text cleaning, etc.
- 'More powerful' algorithms – XGBoost
- More work on tuning model parameters