Introduction to Data Science
CSC59970 | CCNY
Professor Grant M. Long

Final Project
Initial Submission

Abdur Rafey
Dzhonibek Parmankulov
Hasibul Islam

(i) Expected Performance of the Model

To attain a formidable mean squared error for the rents of New York City apartments posted on StreetEasy, we intended to use a couple of models to initialize our tests. The first model we thought of using was Linear Regression.

Mean Squared Error for **Test1** using <u>Linear Regression</u>: 3313817.143868871

The feature columns that we used to test this method were:

> feature_cols = ['bedrooms', 'year_built', 'bathrooms', 'min_to_subway', 'size_sqft', 'no_fee', 'has_doorman', 'Addr_zip', 'floor_count', 'has_gym', 'allows_pets',]

We decided to eliminate the following features as they were beginning to increase the overall mean squared error:

> ['has_elevator', 'has_dishwasher', 'is_furnished', 'has_garage', 'has_pool', 'has_garden']

Although this method provided an acceptable error, it was still not efficient enough as we want to deviate as far left of the 4.0 mark as we can. Next, we tried using the Random Forest Regressor and we obtained the following results:

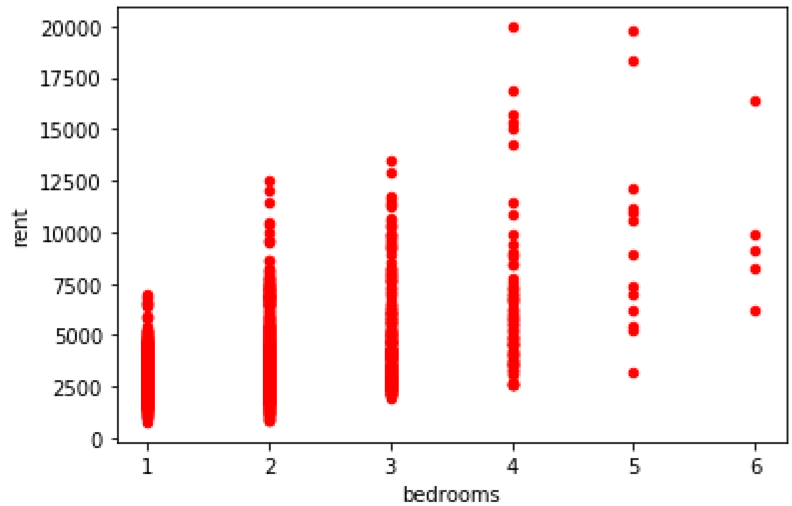
Mean Squared Error for **Test1** using <u>Random Forest Regressor</u>: 1851745.5923584981

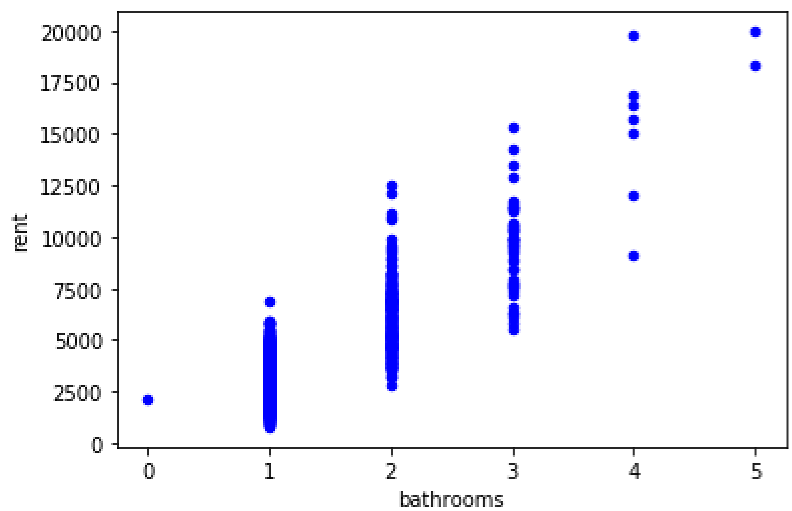
Thus, we were able to attain a much better result using the random forest method. This could be due to the large number of features that we taking into account. Generally, Linear Regression models can be useful for smaller ranges of continuous data. In this case, however, even though we are taking into account only one city, we have many features to handle, including numerous binary variables such as ‘has_doorman’ and ‘has_elevator’. Thus, as we move from one borough to another, more data is fed in and a random forest can handle messier data more efficiently than the regression model. Another hidden factor that could be playing a role in this could be the fact that Linear Regressions require normalization to avoid overfitting, whereas Random Forest has this built-in.

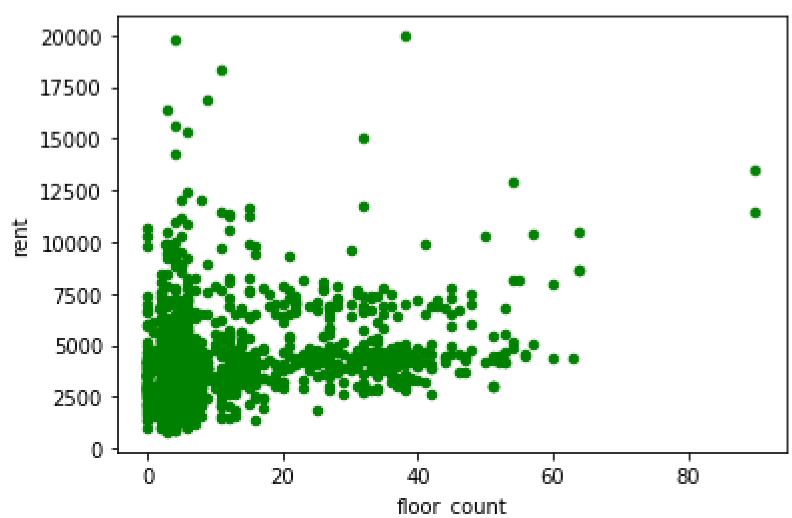
(ii) Intended Strategy to improve the predictions

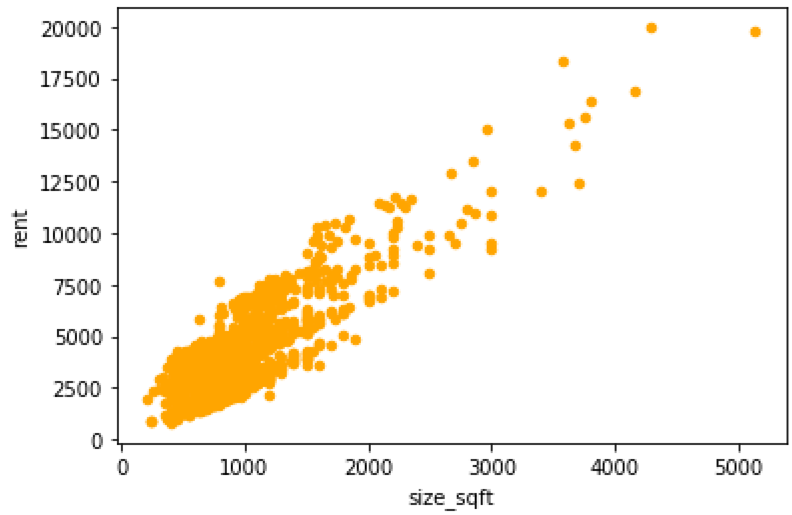
For the final submission, we intend to improve our results by using more models for predicting larger datasets because we intend to add some additional data and functions. We are planning to include popular restaurant spots or grocery shopping locations and see if those variables can be used to predict a higher or lower rent for each location. One of the algorithms we intend to feed our data to is the k-nearest neighbors algorithm. However, since this algorithm can use both classification data and continuous data for regressions, we may fit the variables or separate them in such a way so that we run a separate compilation for classification variables and leave the continuous variables for a regression format of this algorithm. As we use this algorithm against test 3, we will be looking to see how the continuous variables are being predicted compared to our Linear Regression results. In this manner, we can fairly compare each variable with its equal counterpart and see how the classification, binary variables affect the total rental prediction output and the mean squared error with each test run. Afterwards, we will follow the same sequence of steps as our run against test 2 and generate tables and plots of our concluding results.

Graphs for training data:

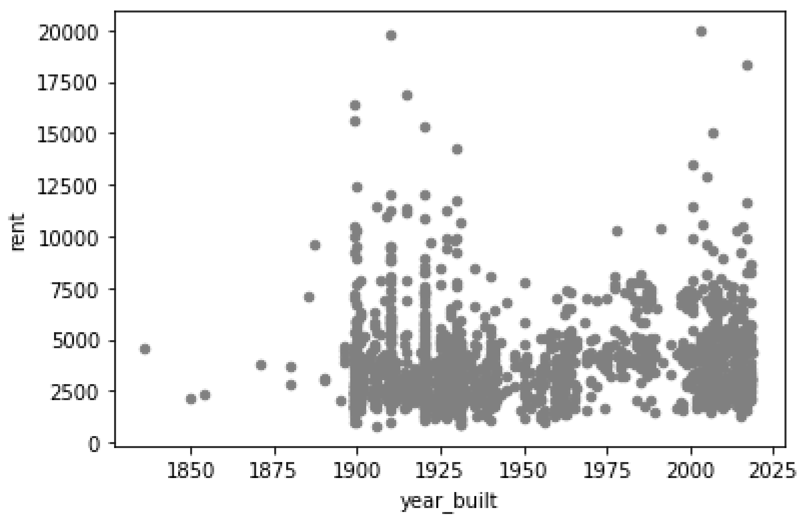




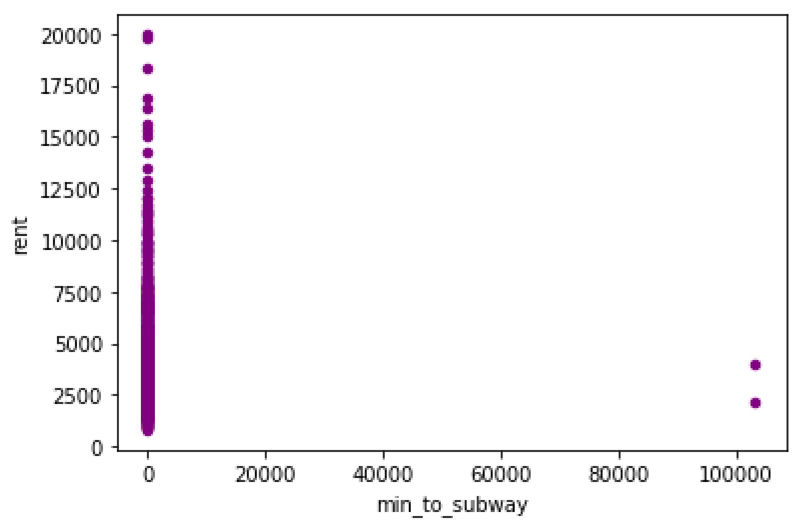




rent
size_sqft

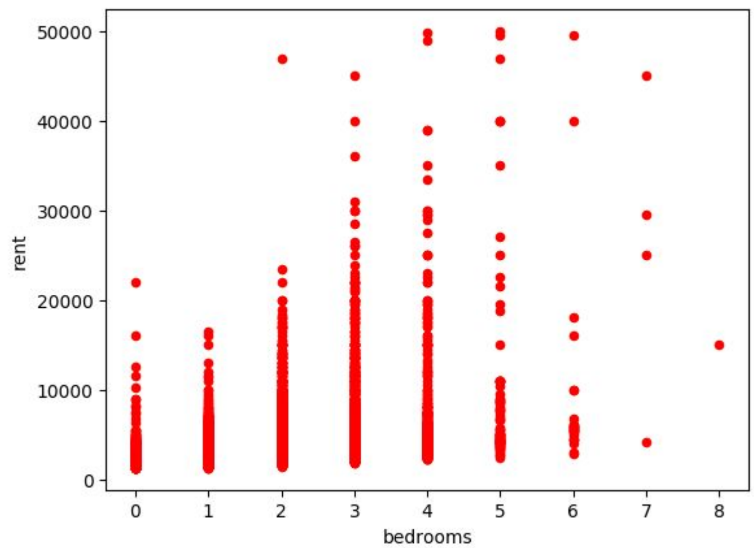


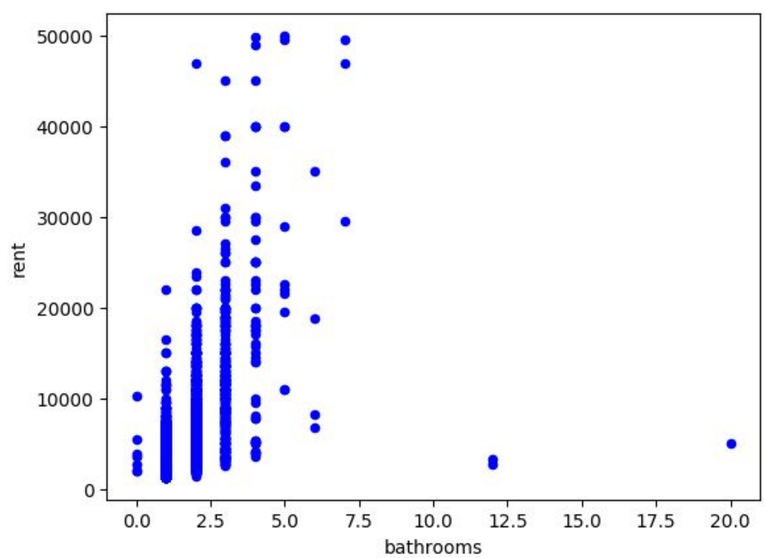
rent
year_built

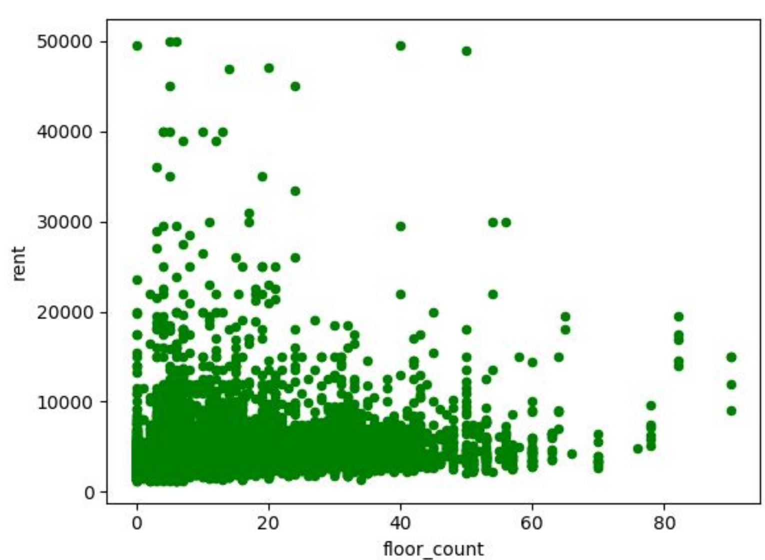


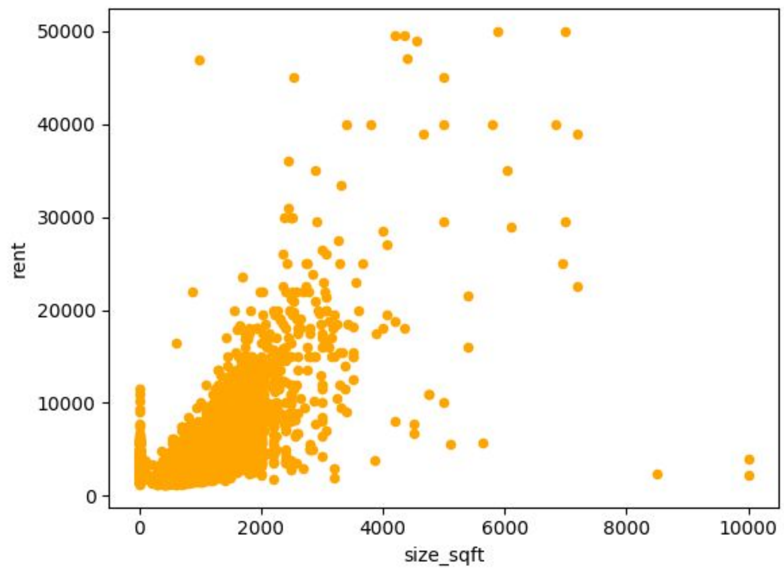
rent
min_to_subway

Graphs for test 2 data:

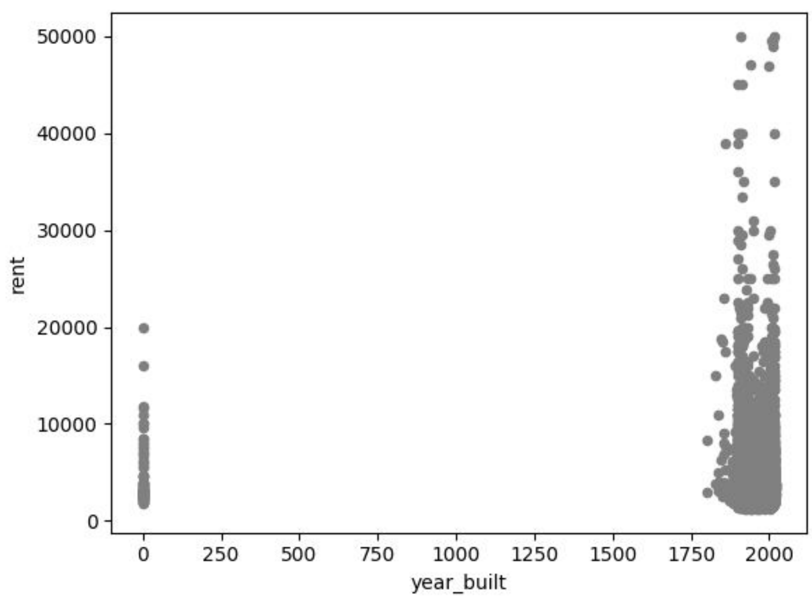




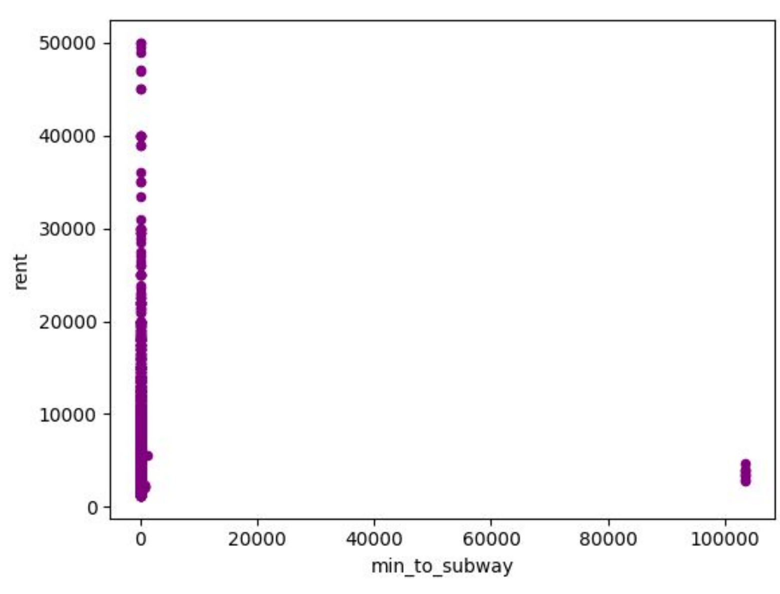




rent
size_sqft



rent
year_built



rent
min_to_subway

OLS Regression Results for Test2 Data:

OLS Regression Results

Dep. Variable:	rent	R-squared:	0.898
Model:	OLS	Adj. R-squared:	0.897
Method:	Least Squares	F-statistic:	1588.
Date:	Wed, 20 Nov 2019	Prob (F-statistic):	0.00
Time:	17:42:28	Log-Likelihood:	-17318.
No. Observations:	2000	AIC:	3.466e+04
Df Residuals:	1989	BIC:	3.472e+04
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
bedrooms	199.7667	37.691	5.300	0.000	125.850	273.684
year_built	2.4586	0.233	10.531	0.000	2.001	2.916
bathrooms	1771.4711	82.298	21.525	0.000	1610.072	1932.870
min_to_subway	0.0149	0.010	1.517	0.129	-0.004	0.034
size_sqft	2.2056	0.085	25.983	0.000	2.039	2.372
no_fee	-270.4860	69.230	-3.907	0.000	-406.257	-134.715
has_doorman	385.0043	112.572	3.420	0.001	164.233	605.776
addr_zip	-0.5609	0.041	-13.525	0.000	-0.642	-0.480
floor_count	21.5113	3.794	5.669	0.000	14.070	28.953
has_gym	482.2676	111.959	4.308	0.000	262.697	701.838
allows_pets	186.3280	68.165	2.733	0.006	52.645	320.011

Omnibus:	1397.915	Durbin-Watson:	2.089
Prob(Omnibus):	0.000	Jarque-Bera (JB):	46484.783
Skew:	2.845	Prob(JB):	0.00
Kurtosis:	25.922	Cond. No.	4.79e+04

OLS Regression Results for Training Data:

OLS Regression Results

Dep. Variable:	rent	R-squared:	0.836
Model:	OLS	Adj. R-squared:	0.836
Method:	Least Squares	F-statistic:	6487.
Date:	Wed, 20 Nov 2019	Prob (F-statistic):	0.00
Time:	17:42:28	Log-Likelihood:	-1.2528e+05
No. Observations:	14000	AIC:	2.506e+05
Df Residuals:	13989	BIC:	2.507e+05
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
bedrooms	145.6283	19.185	7.591	0.000	108.024	183.233
year_built	1.4243	0.091	15.662	0.000	1.246	1.603
bathrooms	1622.6973	36.597	44.340	0.000	1550.963	1694.432
min_to_subway	-0.0023	0.006	-0.355	0.722	-0.015	0.010
size_sqft	2.4718	0.040	61.379	0.000	2.393	2.551
no_fee	-232.1212	34.133	-6.801	0.000	-299.026	-165.216
has_doorman	569.6259	53.495	10.648	0.000	464.768	674.483
addr_zip	-0.3815	0.016	-23.427	0.000	-0.413	-0.350
floor_count	18.6267	1.911	9.749	0.000	14.882	22.372
has_gym	498.2204	51.740	9.629	0.000	396.802	599.638
allows_pets	297.7480	34.079	8.737	0.000	230.950	364.547

Omnibus:	13693.453	Durbin-Watson:	2.023
Prob(Omnibus):	0.000	Jarque-Bera (JB):	3379523.106
Skew:	4.200	Prob(JB):	0.00
Kurtosis:	78.650	Cond. No.	4.32e+04