Machine Learning: Rental Housing Data

...

By Valerie Andrade

Project Overview

Objective

Explore and predict the rental prices in the private property market using machine learning algorithms on web scraped Craigslist housing data

Presentation Sections

Data Acquisition

Data Exploration & Preprocessing

Machine Learning

What I Used

Language

python

Libraries

pandas

matplotlib

scikits learn

BeautifulSoup

Tools

jupyter

GitHub

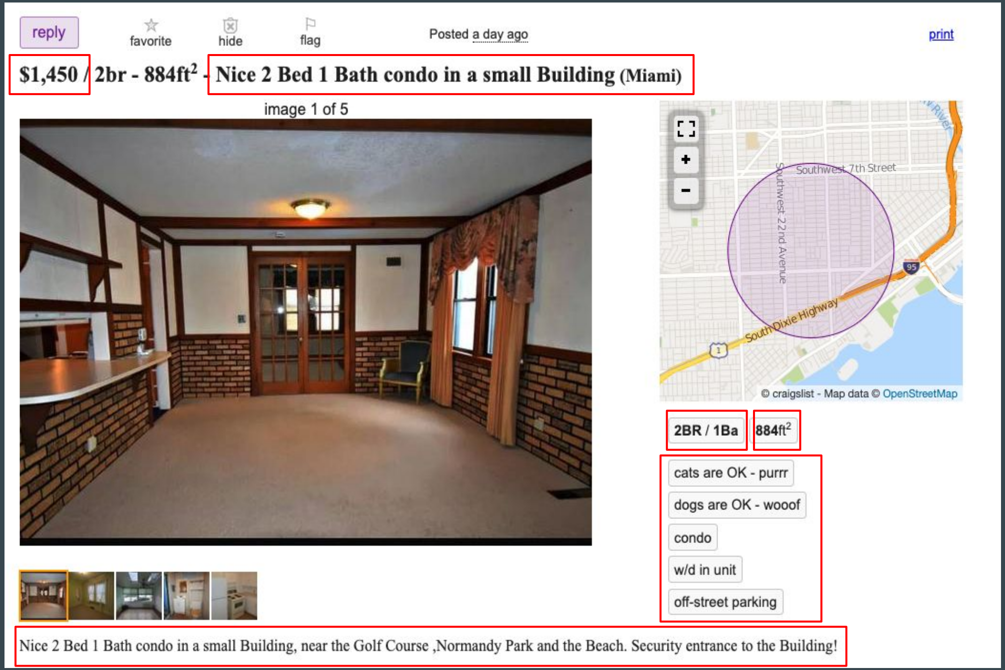
Algorithms

- Linear Regression
- Random Forest

Data Acquisition

Web Scraped Datasets

- Web Scraping code includes a **splinter** that opens each listing (3000) and pulls out the specific details within the post
- Code took about 1.5 hours to webscrape 3000 listings
- I web scraped Miami and San Francisco and used the different data sets on my ML models



Web Scraped DataFrame

	datetimes	hoods	prices	bedrooms	bathrooms	sqft	housing_type	laundry	parking	cats	dogs	furnished
0	2021-05-30 14:02	(Dadeland 2/2)	$1,900	2BR	2Ba	none	apartment	w/d in unit	carport	none	none	none
1	2021-05-30 14:02	(Doral)	$2,557	2BR	2Ba	none	apartment	w/d in unit	attached garage	none	none	none
2	2021-05-30 14:02	(Miami)	$2,241	2BR	2Ba	1106	apartment	w/d in unit	off-street parking	yes	yes	none
3	2021-05-30 14:02	(Doral)	$3,399	3BR	2Ba	none	apartment	w/d in unit	attached garage	none	none	none
4	2021-05-30 14:01	(doral)	$1,978	1BR	1Ba	none	apartment	w/d in unit	attached garage	none	none	none

Data Exploration

Data Exploration - ETL Process

Extract

Extract datasets using web scrape method ensuring it has all necessary information and limited null values.

Null values in sq. ft. presents a challenge

Transform

Transform data by removing duplicate listings, converting data types, replacing string values, creating separate data frames

Convert null values to 0 instead of dropping

Load

Load the data into PostgreSQL

Dataset 1: Miami Rental Listings

neighborhood	br	ba	sqft	housingType	laundry	parking	cats	dogs	furnished	rent
Dadeland	2	2.0	0.0	apartment	w/d in unit	carport	no	no	no	1900
Doral	2	2.0	0.0	apartment	w/d in unit	attached garage	no	no	no	2557
Miami	2	2.0	1106.0	apartment	w/d in unit	off-street parking	yes	yes	no	2241
Doral	3	2.0	0.0	apartment	w/d in unit	attached garage	no	no	no	3399
Doral	1	1.0	0.0	apartment	w/d in unit	attached garage	no	no	no	1978

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1836 entries, 0 to 1835
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   neighborhood  1836 non-null   object
 1   br            1836 non-null   int64
 2   ba            1836 non-null   float64
 3   sqft          1836 non-null   float64
 4   housingType   1836 non-null   object
 5   laundry       1836 non-null   object
 6   parking       1836 non-null   object
 7   cats          1836 non-null   object
 8   dogs          1836 non-null   object
 9   furnished     1836 non-null   object
 10  rent          1836 non-null   int64
dtypes: float64(2), int64(2), object(7)
memory usage: 157.9+ KB
```

	br	ba	sqft	rent
count	1836.000000	1836.00000	1836.000000	1836.000000
mean	1.735294	1.69390	342.716231	2016.338235
std	0.774721	0.55118	538.560115	799.370174
min	0.000000	1.00000	0.000000	500.000000
25%	1.000000	1.00000	0.000000	1500.000000
50%	2.000000	2.00000	0.000000	1800.000000
75%	2.000000	2.00000	750.000000	2250.000000
max	5.000000	4.50000	2921.000000	4950.000000

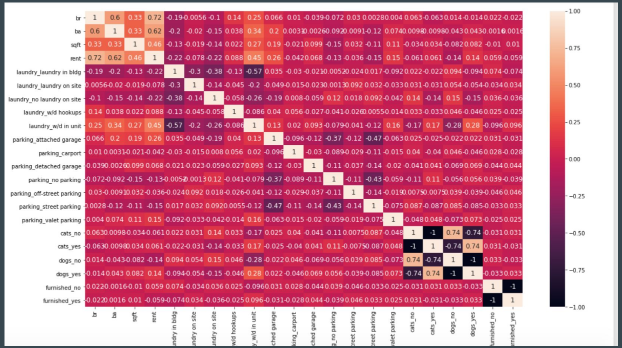
Dataset 2: San Francisco Rental Listings

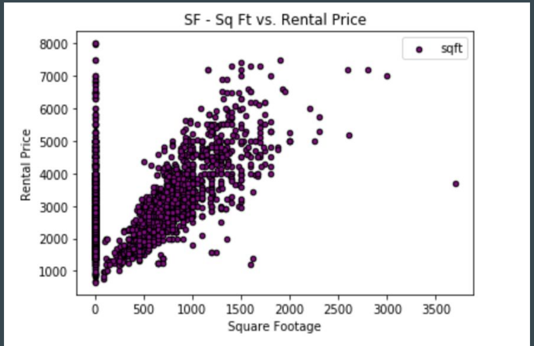
	neighborhood	br	ba	sqft	housingType	laundry	parking	cats	dogs	furnished	rent
	Sunset / Parkside	4	2.0	1600.0	apartment	laundry on site	street parking	no	no	no	4200
	Lower Pac Hts	0	1.0	0.0	apartment	laundry in bldg	attached garage	no	no	no	2850
	SOMA / South Beach	1	1.0	598.0	apartment	w/d in unit	detached garage	yes	yes	no	3431
	Lower Pac Hts	1	1.0	915.0	apartment	w/d in unit	attached garage	no	no	no	3795
	Ingleside / SFSU / CCSF	1	1.0	0.0	apartment	no laundry on site	street parking	no	no	no	2000

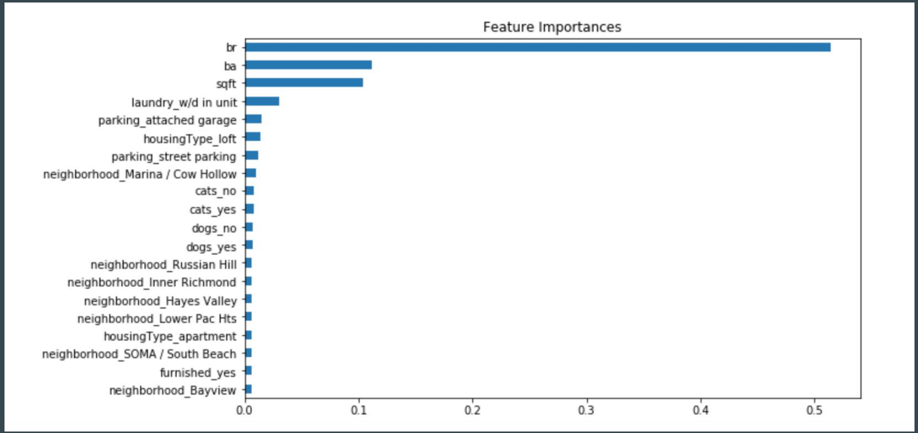
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2596 entries, 0 to 2595
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   neighborhood  2596 non-null   object 
 1   br            2596 non-null   int64  
 2   ba            2596 non-null   float64
 3   sqft          2596 non-null   float64
 4   housingType   2596 non-null   object 
 5   laundry       2596 non-null   object 
 6   parking       2596 non-null   object 
 7   cats          2596 non-null   object 
 8   dogs          2596 non-null   object 
 9   furnished     2596 non-null   object 
 10  rent          2596 non-null   int64  
dtypes: float64(2), int64(2), object(7)
memory usage: 223.2+ KB
```

	br	ba	sqft	rent
count	2596.000000	2596.000000	2596.000000	2596.000000
mean	1.417951	1.211287	402.965331	3007.330123
std	1.095361	0.444560	508.831219	1187.119637
min	0.000000	1.000000	0.000000	650.000000
25%	1.000000	1.000000	0.000000	2190.000000
50%	1.000000	1.000000	0.000000	2795.000000
75%	2.000000	1.000000	750.000000	3688.500000
max	8.000000	4.000000	3700.000000	7995.000000

Data Pre-Processing







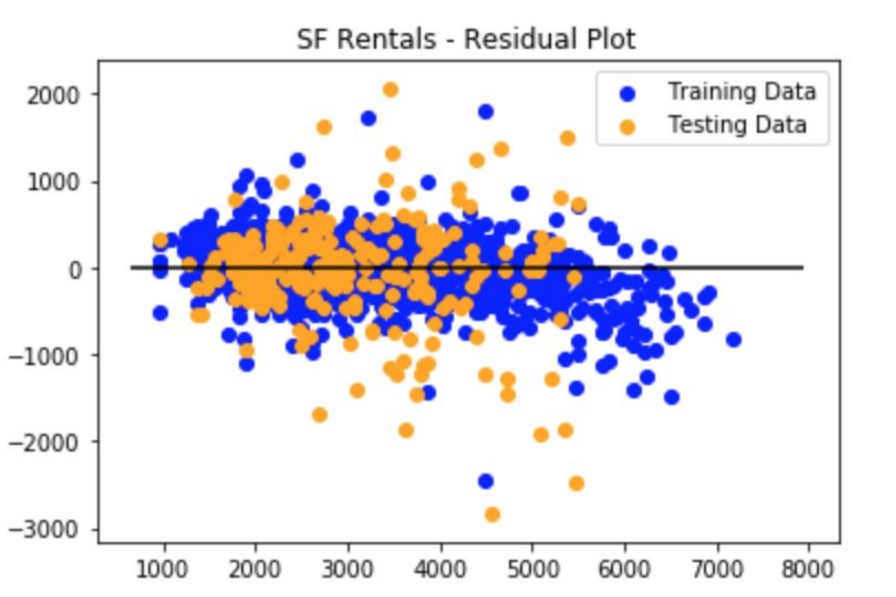
Heatmaps of Correlation Matrix and Scatter Plots to see if there's any obvious relationship between X variables and Rent Price

Machine Learning

Model Testing

Model Name	Data Processing	Model Parameters	Independent Variable(s) Used	RMSE	R2
LinearRegression()	80/20 train test split		sqft	$3,286.06	0.22
LinearRegression() - Multiple	80/20 train test split		br, ba, sqft, housingType, laundry, parking, cats, dogs, furnished	$3,285.83	0.72
RandomForestRegressor()	K-fold cross validation with cv = 10 (10 90/10 splits)	(n_estimators = 1000, random_state = 42, criterion = 'mse', bootstrap=True)	neighborhood, br, ba, sqft, housingType, laundry, parking, cats, dogs, furnished	$579.54	0.7597
RandomForestRegressor()	RandomizedSearch CV param_distribs = {'bootstrap': [True, False], 'max_depth': [10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, None], 'max_features': ['auto', 'sqrt'], 'min_samples_leaf': [1, 2, 4], 'min_samples_split': [2, 5, 10], 'n_estimators': [200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000]}	best_params = 'n_estimators': 600, 'min_samples_split': 5, 'min_samples_leaf': 1, 'max_features': 'sqrt', 'max_depth': 60, 'bootstrap': False	neighborhood, br, ba, sqft, housingType, laundry, parking, cats, dogs, furnished	$567.28	0.7703

Model Findings - Random Forest



	Predicted	Actual	Abs Error
0	1892.201528	2850	957.798472
1	4193.482500	4161	32.482500
2	3711.298333	3165	546.298333
3	2119.985278	1995	124.985278
4	4401.420417	3150	1251.420417
...	...	...	...
254	2753.233750	2295	458.233750
255	1784.722639	1550	234.722639
256	1873.075139	1895	21.924861
257	3446.631250	4595	1148.368750
258	3662.648472	4495	832.351528

Questions?