vacasa

Recommending Similar Rental Units

Talking Points

1. Objectives
2. Exploratory Analysis
3. Data Munging
4. KNN Model Results
5. Conclusion

Objectives

- Develop a programmatic way of determining the “most similar” units to a current unit.
- Maximize conversion/revenue from abandoned cart emails by implementing suggestions of “most similar” units.

Don’t recommend a unit the customer doesn’t want, or worse, can’t stay in!

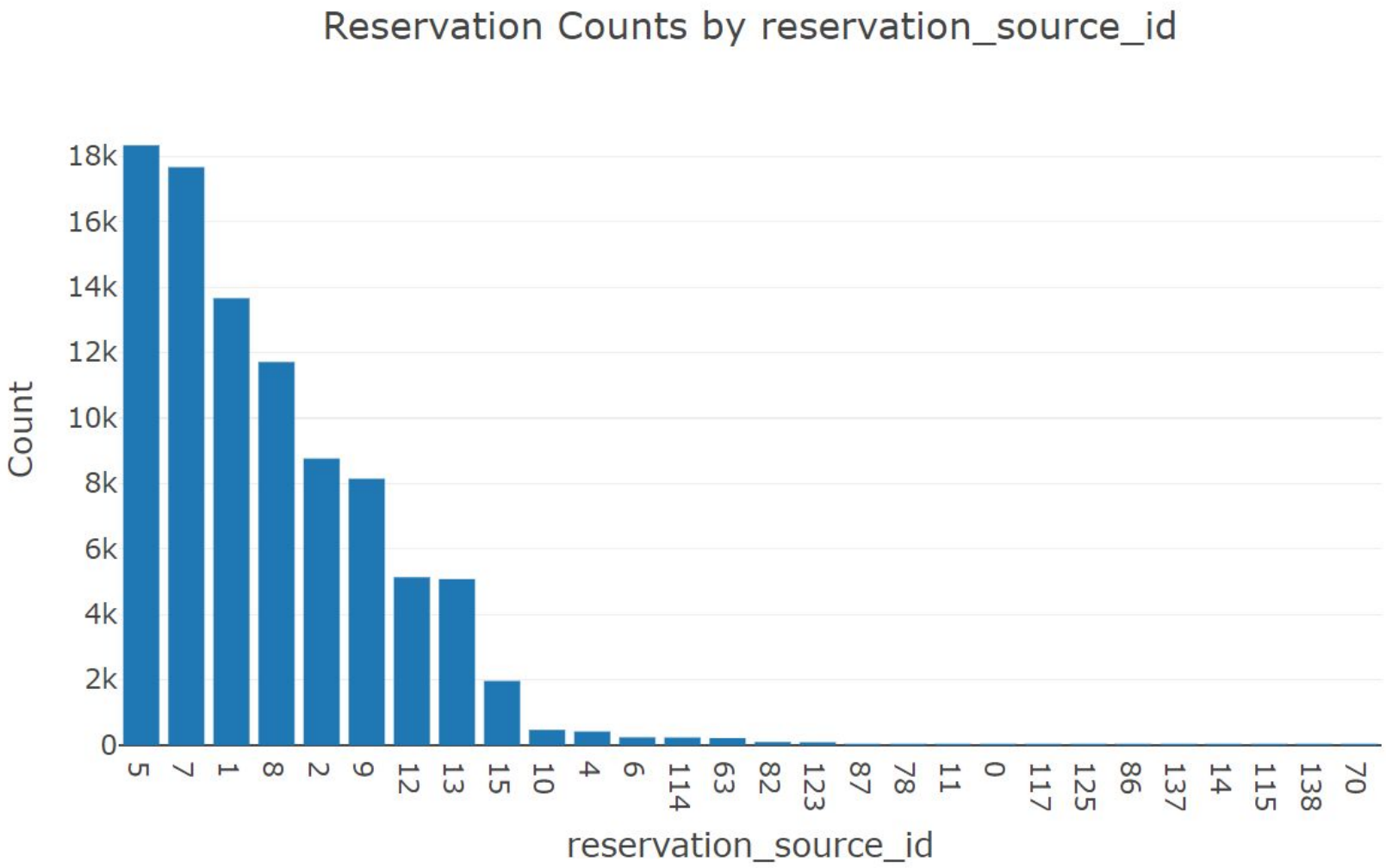
official_reservations

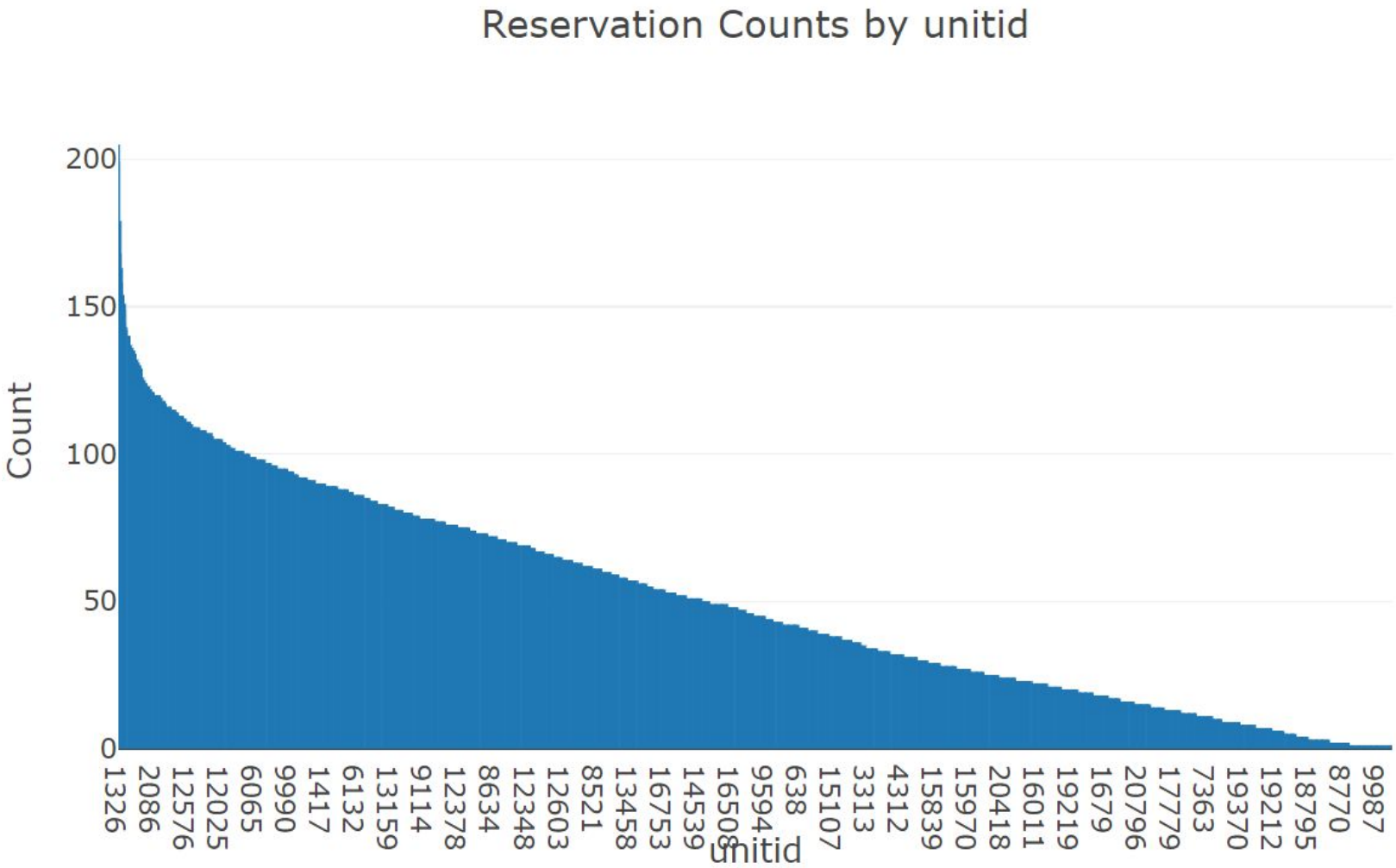
```
First reservation date: 2018-01-01 00:12:25

Last reservation date: 2018-12-31 23:58:41
```

```
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   cancelled              92317 non-null  int64
 1   unitid                 92317 non-null  int64
 2   creationdate           92317 non-null  object
 3   firstnight             92317 non-null  object
 4   lastnight              92317 non-null  object
 5   reservation_source_id  92317 non-null  int64
```

	cancelled	unitid	creationdate	firstnight	lastnight	reservation_source_id
0	0	4703	2018-01-01 08:42:41	2018-02-08 00:00:00	2018-02-11 00:00:00	8
1	1	9980	2018-01-01 19:08:32	2018-04-07 00:00:00	2018-04-13 00:00:00	1
2	0	10938	2018-01-02 01:14:25	2018-01-09 00:00:00	2018-01-10 00:00:00	8
3	0	6057	2018-01-02 10:26:26	2018-07-23 00:00:00	2018-07-25 00:00:00	12
4	1	13154	2018-01-02 09:51:18	2018-01-12 00:00:00	2018-01-14 00:00:00	1


Reservation Counts by reservation_source_id
Count
18k
16k
14k
12k
10k
8k
6k
4k
2k
0
5
7
1
8
2
9
12
13
15
10
4
6
114
63
82
123
87
78
11
0
117
125
86
137
14
115
138
70
reservation_source_id


Reservation Counts by unitid
Count
200
150
100
50
0
1326
2086
12576
12025
6065
9990
1417
6132
13159
9114
12378
8634
12348
12603
8521
13458
16753
14539
16508
9594
638
15107
3313
4312
15839
15970
20418
16011
19219
1679
20796
17779
7363
19370
19212
18795
8770
9987
unitid

units

Look up any unit with the URL:

https://www.vacasa.com/unit/24292

```
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   cityid              2181 non-null   int64
 1   avgbaserate         2181 non-null   int64
 2   dogs                548 non-null    float64
 3   maxoccupancyadults  412 non-null    float64
 4   fullbaths           2181 non-null   int64
 5   terminated          2181 non-null   int64
 6   bedrooms            2176 non-null   float64
 7   beachaccess         465 non-null    object
 8   hottub              2120 non-null   float64
```

unitid	cityid	avgbaserate	dogs	maxoccupancyadults	fullbaths	terminated	bedrooms	beachaccess	hottub
251	41	0	NaN	NaN	2	1	4.0	Drive 5 miles south on 89 to Tahoe City. Afte...	-1.0
536	44	0	NaN	0.0	2	1	5.0	Walk down Ellis to the Lake.	-1.0
876	84	0	NaN	NaN	3	1	4.0	0	1.0
1332	130	0	1.0	NaN	4	1	4.0	NaN	1.0
1374	43	0	NaN	NaN	2	1	3.0	NaN	NaN

https://www.vacasa.com/unit/24292

Select Your Dates | 1 guest | $74-$371 per night | Book

USA › California › Groveland › Pine Mountain Lake › Listing #24292

Green Valley Hideaway (370/03) - Groveland, CA

Save to Favorites | Take a Virtual Tour

10 reviews

Max. occupancy: 5 | 1 Queen Bed, 1 Double Bed, 1 Twin Bed | 2 bedrooms

2 bathrooms | No pets | 4WD/traction may be required in winter

Important to remember that the data is a snapshot of 2018 and may not match live site data!

AMENITIES

- Hot tub (Private)
- Internet
- Pool (Private)
- Washer/dryer (Private)
- Cable
- Gas fireplace
- Sauna (Private)
- Tennis court (Shared)

FEATURES

Washer/dryer : Private

HEATING & COOLING

Central AC
Fireplace
~~Partial AC~~
Gas fireplace

KITCHEN & DINING

Dishwasher
Fridge
Microwave
Stove

LOCATION

Mountain View

MEDIA

Internet
TV
Cable
Wireless router

NEARBY ACTIVITIES & ATTRACTIONS

Golf on-site
Golf Nearby

ON-SITE ACTIVITIES

Hot tub : Private
Pool : Private
Sauna : Private
Tennis court : Shared

OUTDOOR

Gas grill : Private
Deck

cityid

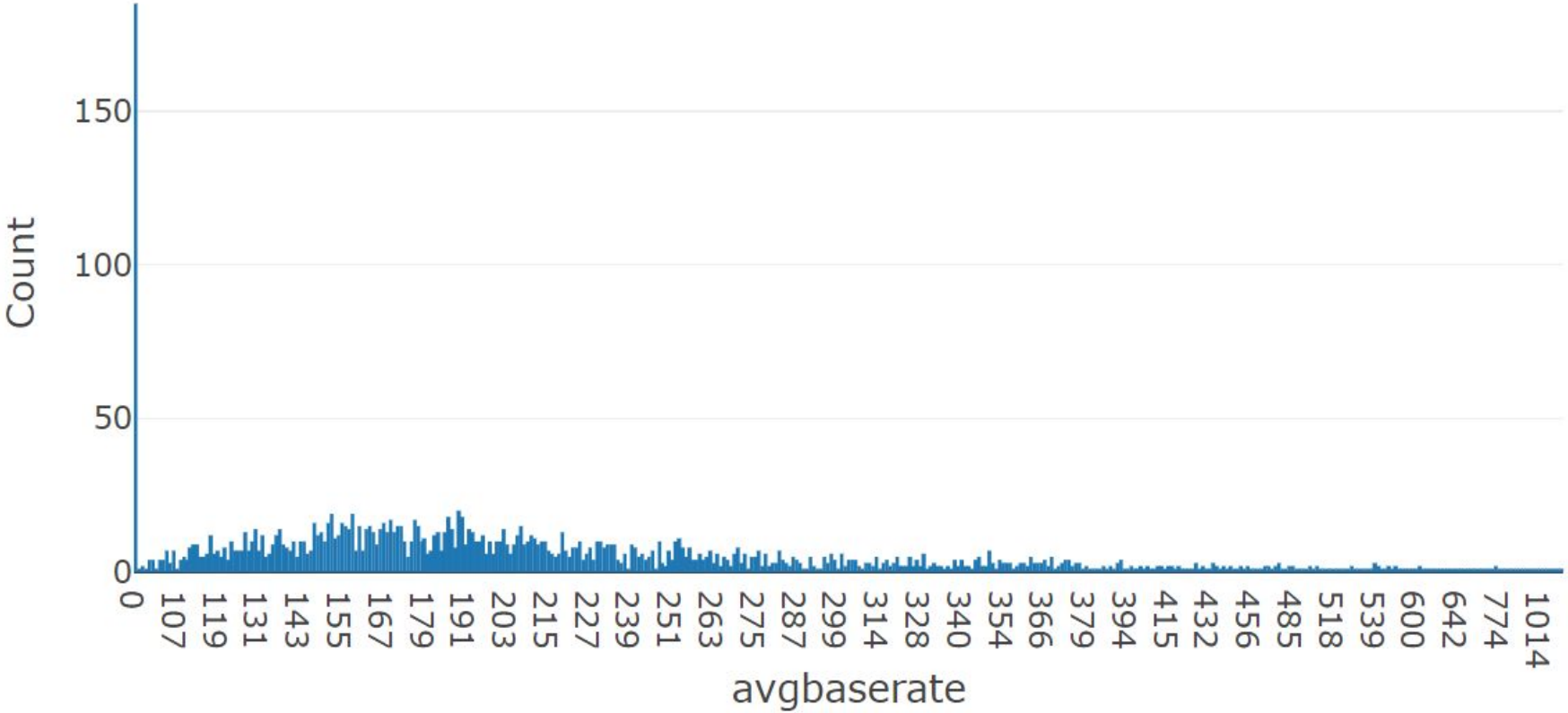
- 52 cities have 3 or less total units.
- 32 cities have only exactly 1 unit.
- It’s unknown if cityid is correlated with city locations.

We need a method to deal with these edge cases...

In production we would have the lat + lon of each unit. We could simply find units in nearby towns/cities.

In this case, we can recommend remaining units in the city for cities with only 2 or 3 units. For cities with only 1 unit, we can suggest another similar or nearby city with more units.

- It’s unclear what an avgbaserate of 0 represents.



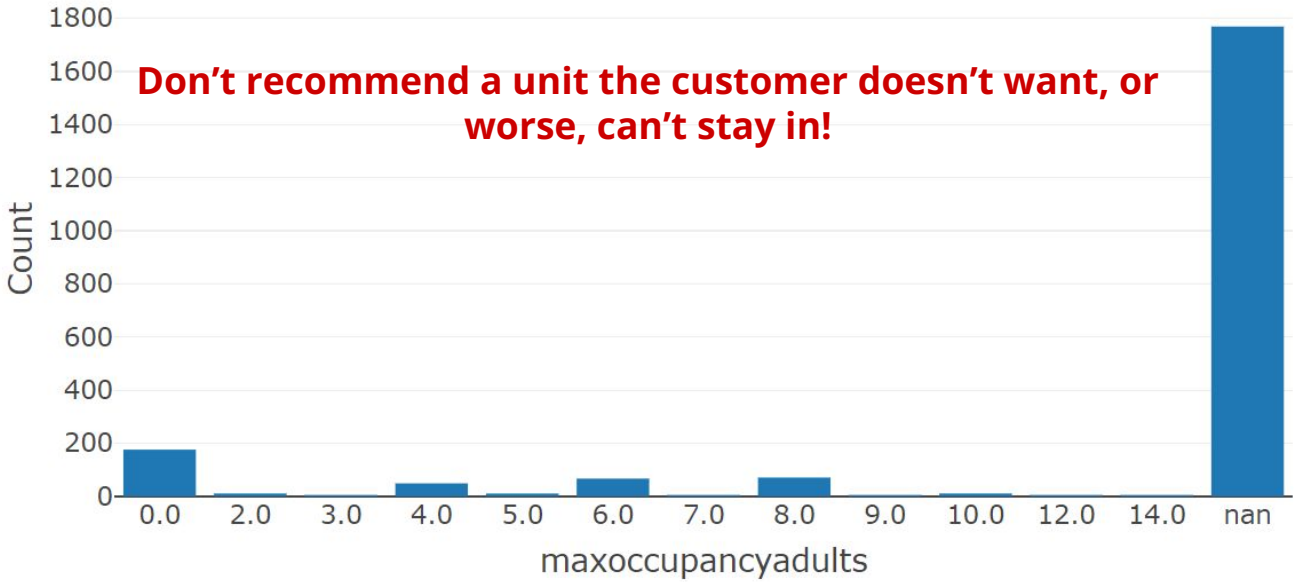
dogs

```
NaN    1633
1.0     529
0.0      19
Name: dogs, dtype: int64
```

- 75% missing values.
- Of the non-missing values, 97% of units allow dogs.
- If customer is reserving unit allowing dogs, they're probably not interested in units that don't allow dogs.
- A customer who isn't searching for a unit that allows dogs however, may not care whether the unit allows them or not.
- Err on the side of caution and fill missing values with 0.

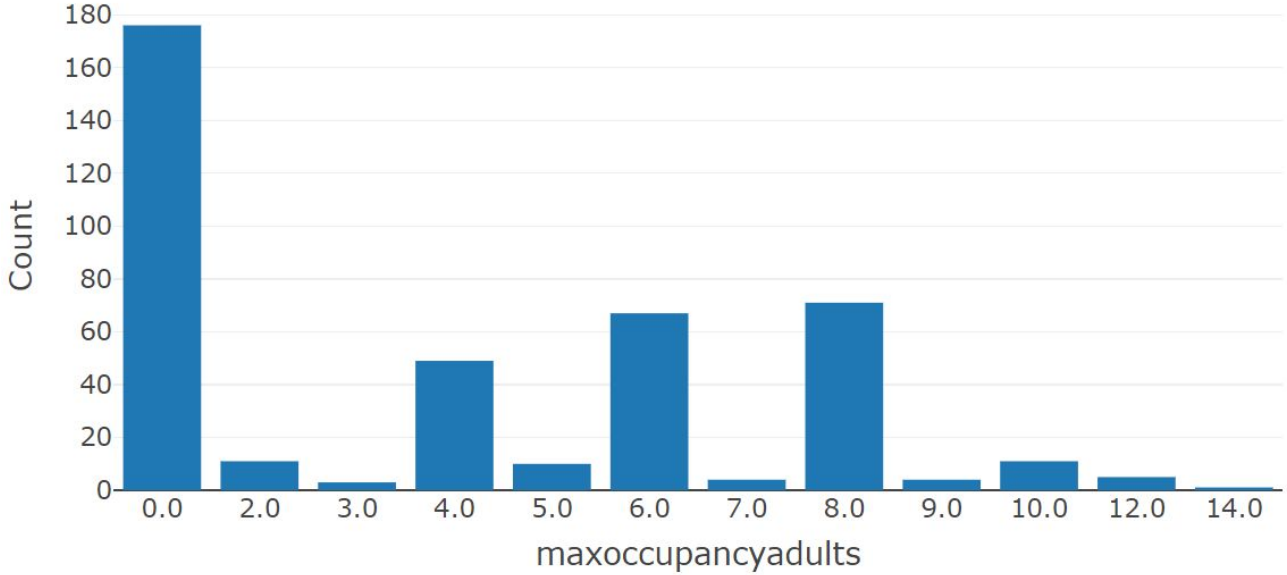
maxoccupancyadults

- 81% missing data.
- Critical feature so we want to impute.

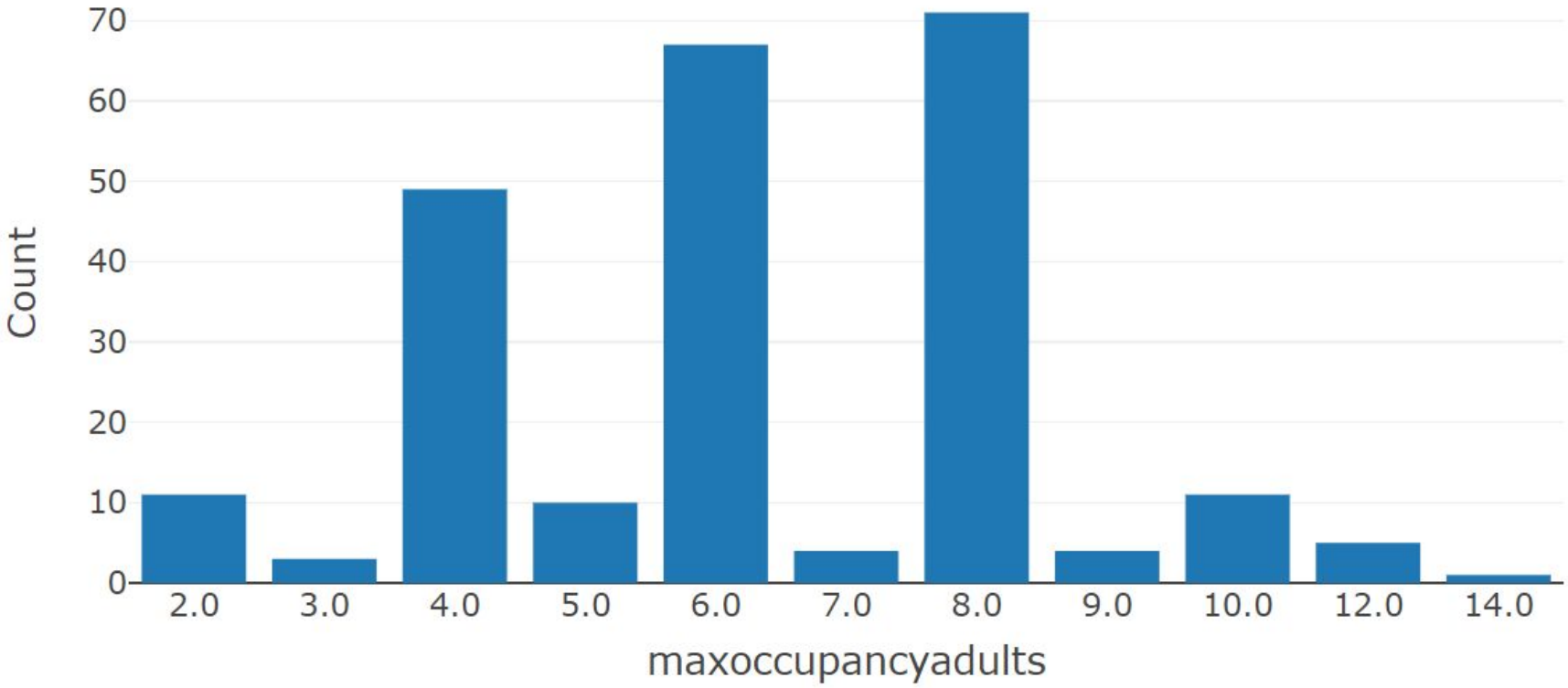


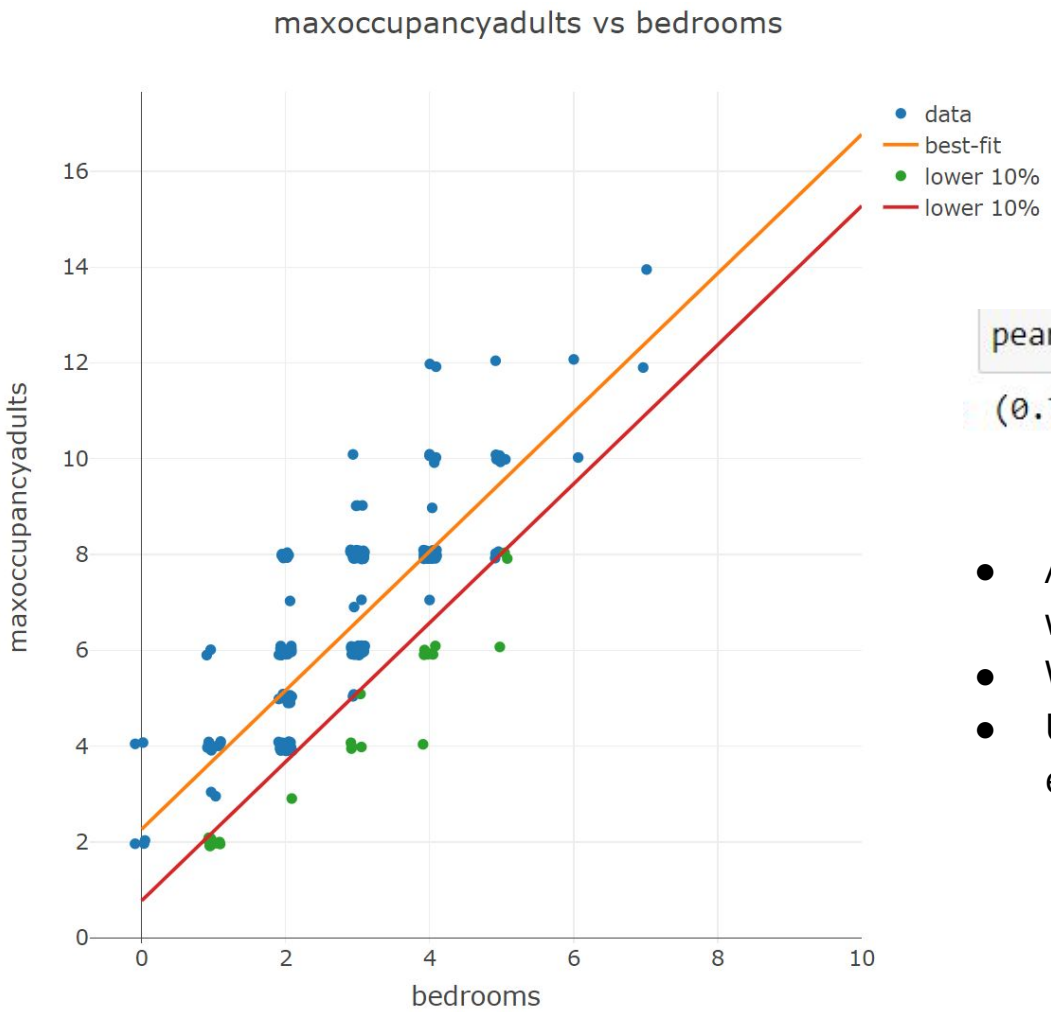
maxoccupancyadults

- Unclear what 0 encodes in *maxoccupancyadults*.
- Need to understand before running model in production.



maxoccupancyadults

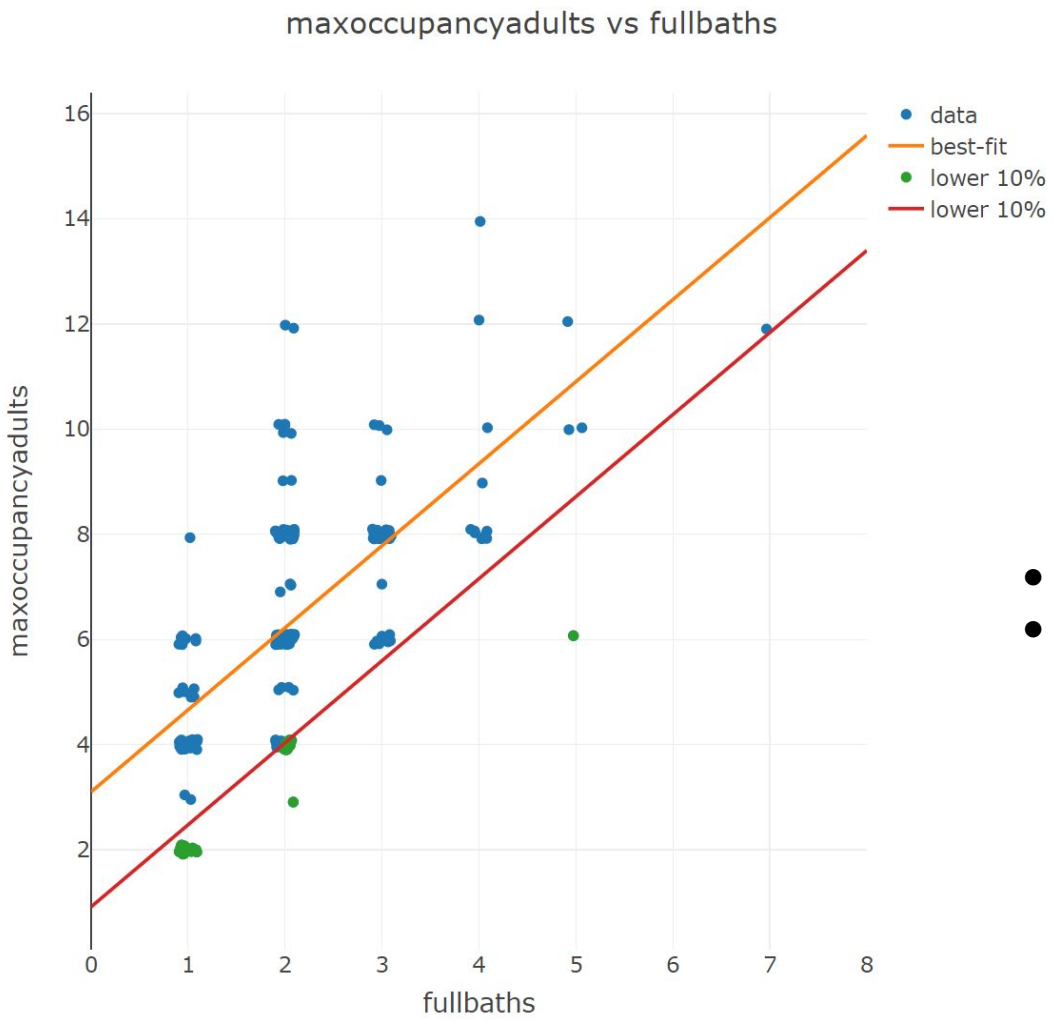





```
pearsonr(d.bedrooms, d.maxoccupancyadults)
```

```
(0.7868459167808943, 5.904764403149468e-51)
```

- Again, to err on the side of caution we want to avoid imputing using the best-fit.
- We can use a lower percentile instead.
- Using the minimum max occupancy for each bedroom would be the safest.

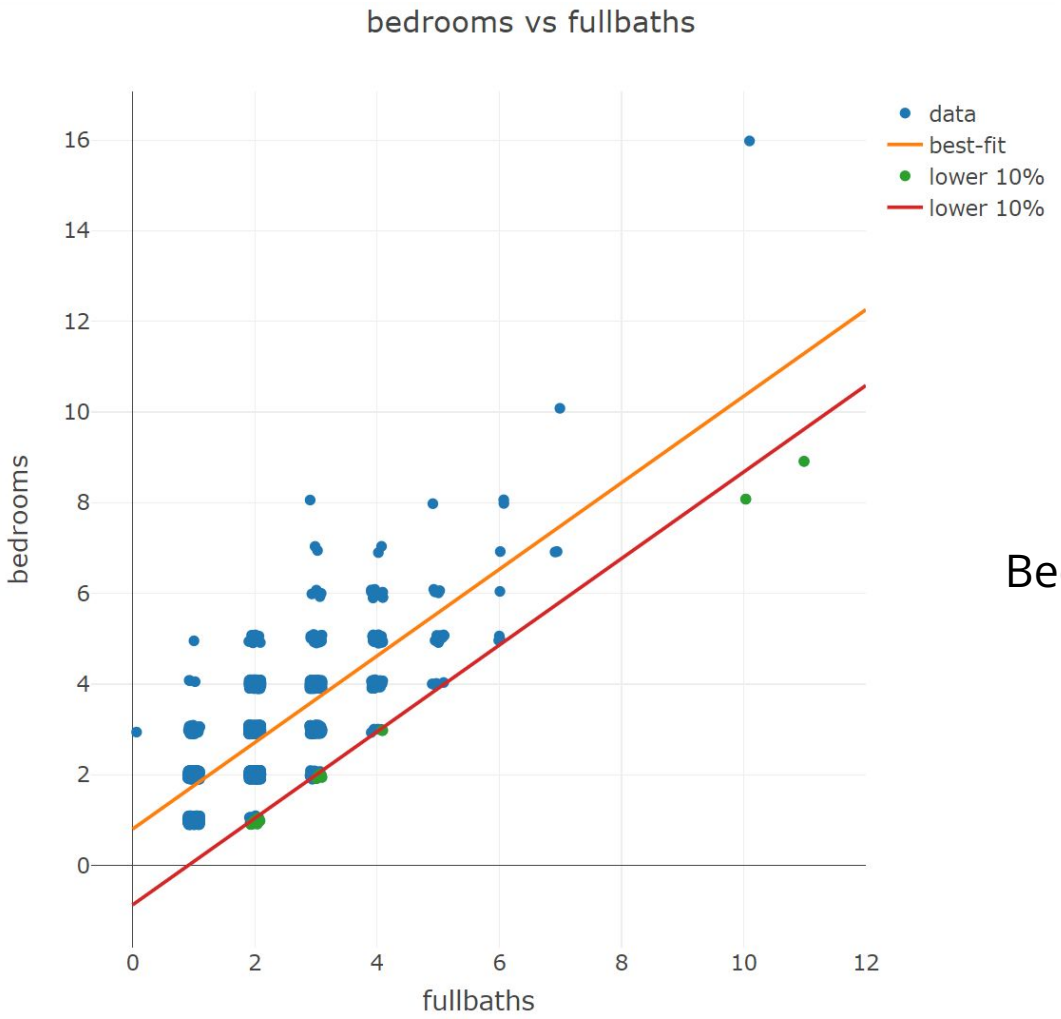


- Bedrooms itself has missing values.
- I use a similar approach to fill the rest using fullbaths.

terminated

- No missing values.
- Unclear what terminated is so dropped until clearer understanding.

	cancelled	unitid	creationdate	firstnight	lastnight	reservation_source_id	cityid	avgbaserate	dogs	maxoccupancyadults	fullbaths	terminated	bedrooms
0	0	4703	2018-01-01 08:42:41	2018-02-08 00:00:00	2018-02-11 00:00:00	8	41.0	439.0	NaN	NaN	4.0	1.0	5.0
1	1	9980	2018-01-01 19:08:32	2018-04-07 00:00:00	2018-04-13 00:00:00	1	370.0	162.0	1.0	2.0	1.0	1.0	1.0
2	0	10938	2018-01-02 01:14:25	2018-01-09 00:00:00	2018-01-10 00:00:00	8	46.0	119.0	NaN	NaN	1.0	0.0	0.0
3	0	6057	2018-01-02 10:26:26	2018-07-23 00:00:00	2018-07-25 00:00:00	12	545.0	163.0	NaN	NaN	2.0	0.0	2.0
4	1	13154	2018-01-02 09:51:18	2018-01-12 00:00:00	2018-01-14 00:00:00	1	32.0	158.0	NaN	NaN	2.0	1.0	3.0



Bedrooms imputed from fullbaths

beachaccess

- 79% missing.
- Non-missing values mostly directions to access beach.
- Encode 1 for beach access and 0 for no access.

unitid	cityid	avgbaserate	dogs	maxoccupancyadults	fullbaths	terminated	bedrooms	beachaccess	hottub
251	41	0	NaN	NaN	2	1	4.0	Drive 5 miles south on 89 to Tahoe City. Afte...	-1.0
536	44	0	NaN	0.0	2	1	5.0	Walk down Ellis to the Lake.	-1.0
876	84	0	NaN	NaN	3	1	4.0	0	1.0
2232	225	138	1.0	0.0	2	1	3.0	Lake Elsinore is 10 minutes from home. \r\nFor...	-1.0
3277	368	106	NaN	0.0	1	1	1.0	One block walk to the ocean and public beach. ...	2.0

hottub

Original data

- -1: no hot tub
- 1: hot tub (private)
- 2: hot tub (shared)

Cleaned data

- Change -1 to 0
- Fill missing with 0

```
-1.0    1209
 2.0     471
 1.0     440
 NaN      61
Name: hottub, dtype: int64
```

segment_page_views

- Neither anonymous_id or unit_id are unique.
- Can groupby unitid to get page views for each unit.

```
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   anonymous_id  96845 non-null  object
 1   unit_id       96845 non-null  int64
```

	anonymous_id	unit_id
0	68ebf758-2bb1-4719-a463-6e15540f0a08	11774
1	68ebf758-2bb1-4719-a463-6e15540f0a08	959
2	04041e02-7a3e-4d62-b862-610234b47943	11818
3	04041e02-7a3e-4d62-b862-610234b47943	11818
4	70dce651-7631-44e0-b677-c286dd57fb1c	12347

segment_reservations

- Anonymous_id and reservation_id not unique.
- No reservation_id in official_reservations to join.
- Can’t join to segment_page_views.
- If we have anonymous_id along with abandoned cart unit, could use customer history to make more personalized recommendations.

```
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   anonymous_id    4830 non-null   object
 1   reservation_id  4830 non-null   int64
```

	anonymous_id	reservation_id
0	cb4d14ab-ee3d-431d-aebd-2b46651dd280	1167636
1	726448eb-9ffc-4c7b-af00-5b0ac02c3ae6	1172722
2	4f753e6d-a71c-4196-a600-a2e01b826b94	1174695
3	16a8ebe3-fbec-4f75-a452-4e83ef64c735	1178629
4	ab0c1b24-e828-4a66-92d4-8bcc69c5ddca	1183434

Normalization

- Normalize to 0 mean and unit variance.

```
scaler = StandardScaler()

scaler.fit(units_no_city)

units_transformed = pd.DataFrame(scaler.transform(units_no_city), columns=units_no_city.columns, index=units_no_city.index)
units_transformed.head()
```

unitid	avgbaserate	dogs	maxoccupancyadults	fullbaths	bedrooms	beachaccess	hottub
251	-1.538004	-0.565878	1.057118	-0.066570	1.025093	1.934245	-0.777593
536	-1.538004	-0.565878	-1.989706	-0.066570	1.832112	1.934245	-0.777593
876	-1.538004	-0.565878	1.057118	1.016938	1.025093	1.934245	0.449564
1332	-1.538004	1.767165	1.057118	2.100446	1.025093	-0.516998	0.449564
1374	-1.538004	-0.565878	0.186597	-0.066570	0.218073	-0.516998	-0.777593

Data Munging Summary

1. Impute dogs with 0.
2. Impute maxoccupancyadults from bedrooms, then fullbaths.
3. Impute bedrooms from fullbaths.
4. Drop *terminated* feature.
5. Impute *beachaccess* with 0
6. Replace -1 with 0 in hottub and impute with 0.
7.

KNN Model

- Get random unit from units table.
- Apply data munging & normalization.
- Subset data to units within same city.
- NearestNeighbor() from sklearn.

Example 1

```
# choose random unit
sample = units.sample()
sample
```

unitid	cityid	avgbaserate	dogs	maxoccupancyadults	fullbaths	terminated	bedrooms	beachaccess	hottub
1674	32	379	NaN	NaN	3	0	6.0	The West End Beach Donner Lake State Park is d...	-1.0

```
# run random unit through model
top5 = model.run_KNN(sample, n_neighbors=5, n_samples=4)
top5
```

unitid	cityid	avgbaserate	dogs	maxoccupancyadults	fullbaths	terminated	bedrooms	beachaccess	hottub
1674	32	379	NaN	NaN	3	0	6.0	The West End Beach Donner Lake State Park is d...	-1.0
4293	32	429	NaN	NaN	3	1	4.0	Guests can use the private beach and marina at...	1.0
3737	32	283	NaN	NaN	3	1	4.0	Guests can use the private beach and marina at...	1.0
6247	32	219	NaN	NaN	2	0	4.0	Donner Lake is a 5 minute walk. There are 35 p...	-1.0

Example 1 cont.

```
# choose random unit
sample = units.sample()
```

```
units[units.cityid==32].sort_values('maxoccupancyadults', ascending=False)
```

unitid	cityid	avgbaserate	dogs	maxoccupancyadults	fullbaths	terminated	bedrooms	beachaccess	hottub
12492	32	415	NaN	10.0	3	0	5.0	NaN	-1.0
9321	32	212	NaN	9.0	3	0	3.0	Access to Tahoe Donner Beach and Marina at Don...	2.0
16656	32	250	NaN	8.0	3	1	4.0	NaN	2.0
2174	32	364	NaN	8.0	3	0	4.0	Drive to Lake Tahoe on 267-N and park in Kings...	2.0
17592	32	215	NaN	8.0	2	0	3.0	NaN	1.0

4293	32	429	0.0	7.0	3	4.0	1	1.0
3737	32	283	0.0	7.0	3	4.0	1	1.0
6247	32	219	0.0	7.0	2	4.0	1	0.0

Example 2

unitid	cityid	avgbaserate	dogs	maxoccupancyadults	fullbaths	terminated	bedrooms	beachaccess	hottub
16217	464	0	NaN	NaN	3	0	4.0	NaN	1.0
4280	464	158	NaN	NaN	3	0	4.0	NaN	-1.0
17982	464	136	NaN	NaN	2	0	4.0	NaN	-1.0
19594	464	195	NaN	NaN	2	1	3.0	NaN	1.0

unitid	cityid	avgbaserate	dogs	maxoccupancyadults	fullbaths	terminated	bedrooms	beachaccess	hottub	anonymous_id
16217	464	0	NaN	NaN	3	0	4.0	NaN	1.0	NaN
4280	464	158	NaN	NaN	3	0	4.0	NaN	-1.0	56.0
17982	464	136	NaN	NaN	2	0	4.0	NaN	-1.0	97.0
19594	464	195	NaN	NaN	2	1	3.0	NaN	1.0	13.0

Learnings

1. Some features such as maxoccupancyadults may have to be weighted more than others such as avgbaserate for example.
2. More work on data collection/integrity in the frontend can prevent a lot of error/bias in the models.
3. Can use data about unit reservations and user history data to increase conversion/revenue.