Airbnb Monthly Rental Capacity Predictions

Why?

- $74.64B in 2021 Market Valuation (Vacation Rentals)
- Expected 5.3% growth from 2022 to 2030
- Can expect even more growth with remote work being more widely acceptable

What

- Help Airbnb hosts to predict the residencies occupancy for given month
- Geocentric based regression models
-

How

- Utilizing data from airdna that aggregates short-term rental analytics
  - Started with Austin Area
- Identify through EDA what features in the data provide valuable insight
- Employ the use of machine learning to provide predictive power to help airbnb hosts (experienced and beginner)

Data Insights

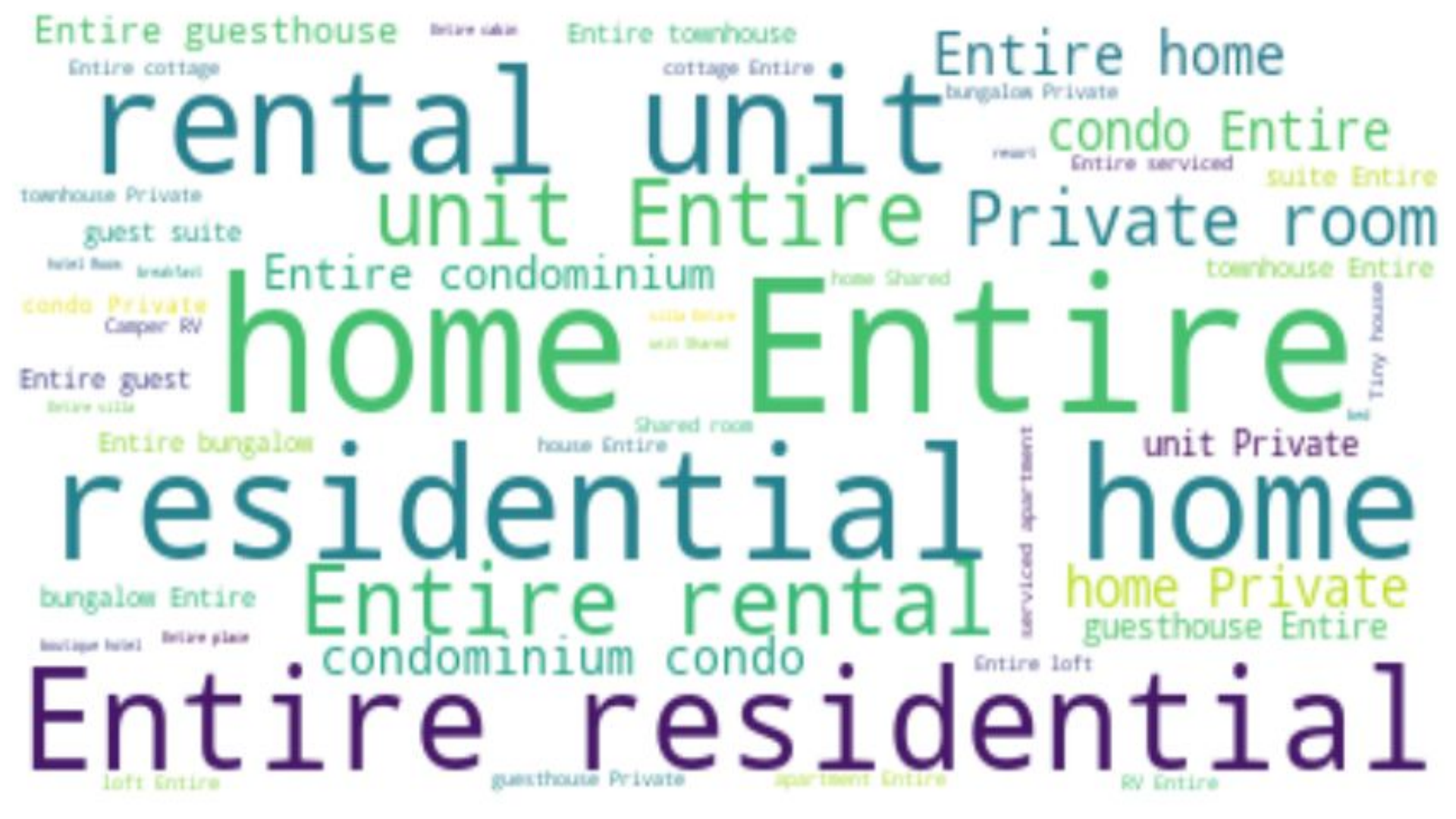
Split the data into two sets (Residences with less than 50% availability and ones with 50% or more availability for the month) and got the following insights:

- Location, location, location
- Amenities (Free parking, air conditioning, long-term stays, etc.)
- Preferred Property Types (Entire spaces)

Data Insights Cont (Categorization)

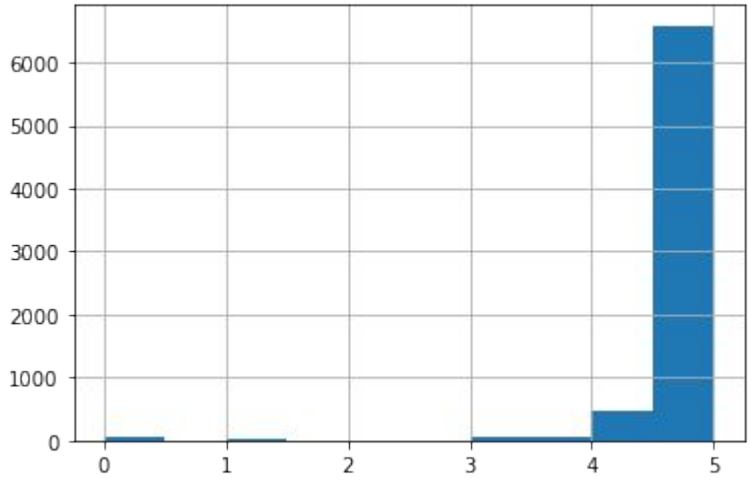
Categorized the following fields:

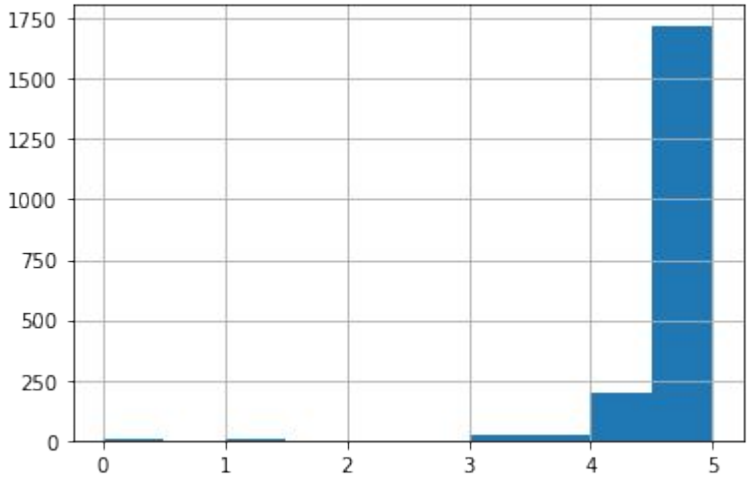
- Neighbourhood
- Property Type
- Top Amenities


Entire guesthouse
Entire townhouse
Entire home
Entire cottage
cottage Entire
rental unit
bungalow Private
condo Entire
Entire serviced
suite Entire
townhouse Private
unit Entire
Private room
guest suite
Entire condominium
townhouse Entire
home Shared
home Entire
condo Private
Camper RV
Tiny house
Entire guest
Shared room
Entire bungalow
house Entire
unit Private
residential home
serviced apartment
home Private
Entire rental
bungalow Entire
guesthouse Entire
condominium condo
Entire loft
Entire residential
loft Entire
guesthouse Private
apartment Entire
RV Entire

Data Insights Cont.

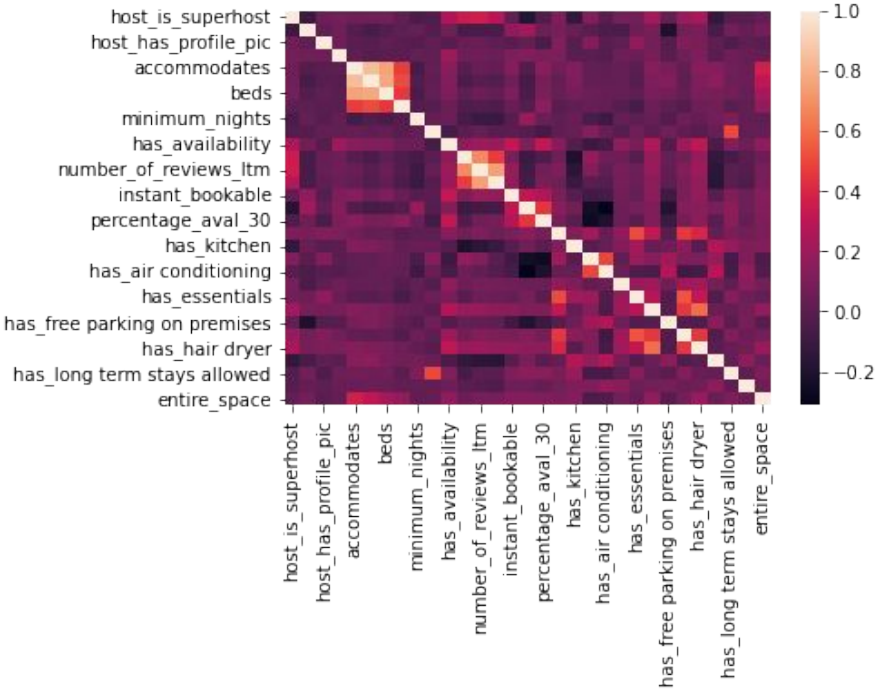
Based on the two groups mentioned earlier there were similarities between their overall ratings:





Data Insights Cont. (Correlations)

- Identify correlations between independent features and our target variable



Pre-Modeling (Standardizing Data)

- Ensure that data types are numeric values (no strings)
- Split data into train and test sets
- Standardize (scale) feature input for model

Modeling the Data

For this project the following Models were utilized:

- Dummy Regression
- Linear Regression
- Lasso
- Ridge Regression

Modeling the Data cont. (Optimizing ML Model)

Used a Dummy Regression model as a baseline to compare my other models to specifically accuracy and variance. Which raised three questions to be answered:

- Can we reduce number of features?
- Can we improve our models ability to handle variance? (r2 score)?
- Can we improve overall accuracy of model? (mean absolute error)

Modeling the Data cont. (Streamlining ML model building)

Sklearn provides libraries that allow us to streamline our creation of models and various parameters

```
lasso_model = Lasso()

#create pipeline
pipe_lasso = make_pipeline(
    SimpleImputer(strategy='mean'),
    StandardScaler(),
    SelectKBest(f_regression),
    lasso_model,
)
```

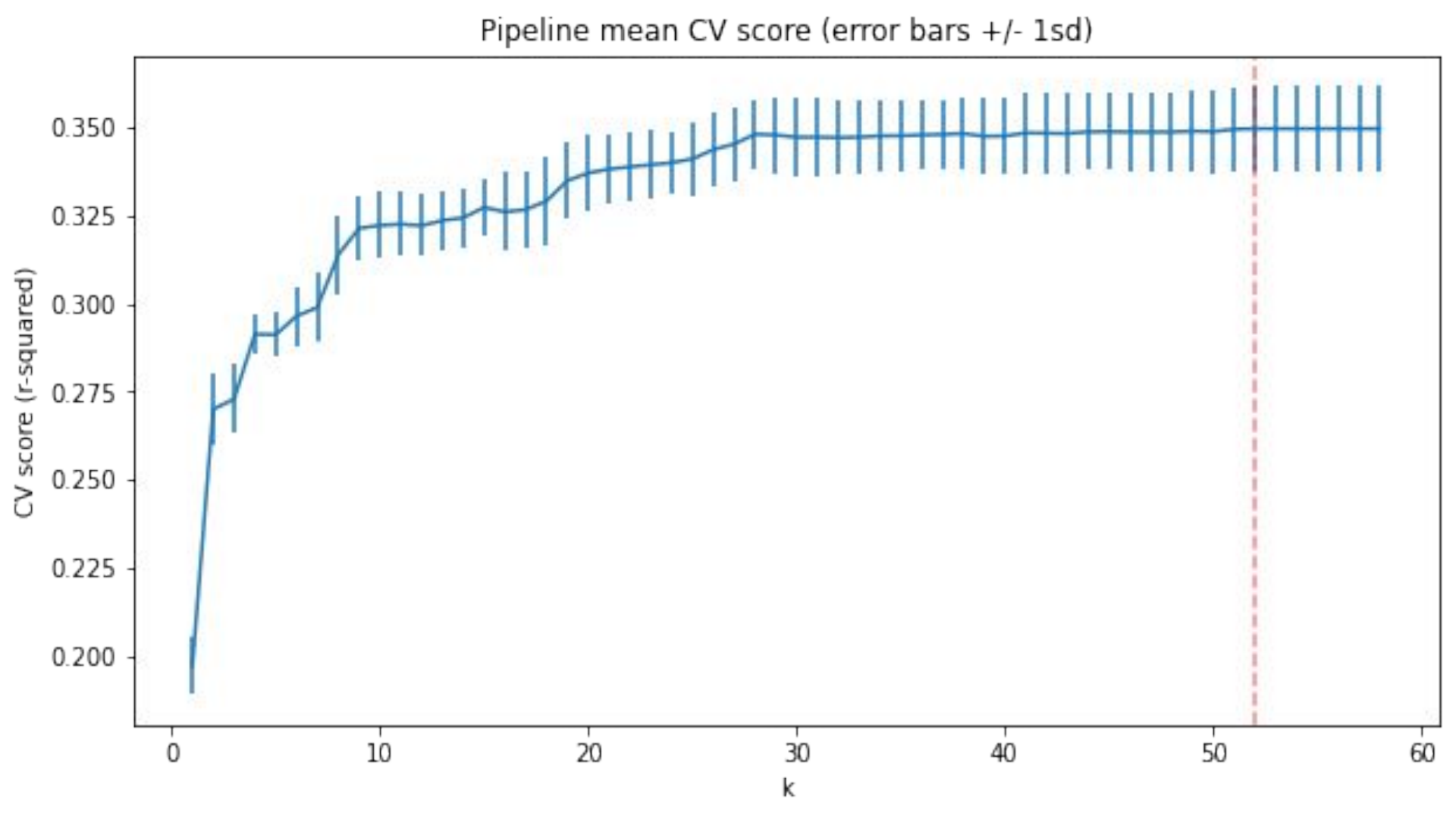
Model the Data cont. (Optimizing Hyperparameters)

- GridSearchCV
  - Regularization of parameters for Lasso and Ridge Regression
  - Selection of optimal features to use for training the model

```
grid_params = {'lasso__alpha': np.linspace(0, 0.01, 10),
'selectkbest__k': np.arange(0, X_train.shape[1])*1+1}

lasso_grid_cv = GridSearchCV(pipe_lasso, param_grid=grid_params,
cv=5, n_jobs=-1)

lasso_grid_cv.fit(X_train, y_train)
```


Pipeline mean CV score (error bars +/- 1sd)
CV score (r-squared)
0.350
0.325
0.300
0.275
0.250
0.225
0.200
0
10
20
30
40
50
60
k

Metrics

Model Name	R2 Score (train)	R2 Score (test)	MAE (test)	MAE (train)	Features
Dummy Regression	0	-9.537732032 22763E-05	0.28	0.27	58
Linear Regression	0.36	0.37	0.2	0.2	58
Laso Regression	0.36	0.37	0.2	0.2	52
Ridge Regression	0.36	0.37	0.2	0.2	52

Conclusion

- Ridge Regression model was selected for:
  - Highest r2 score
  - Lower mean absolute error
  - Use of less features (52 out of 58)