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 if there residence will have monthly booking of 50%
- Geocentric based classification models

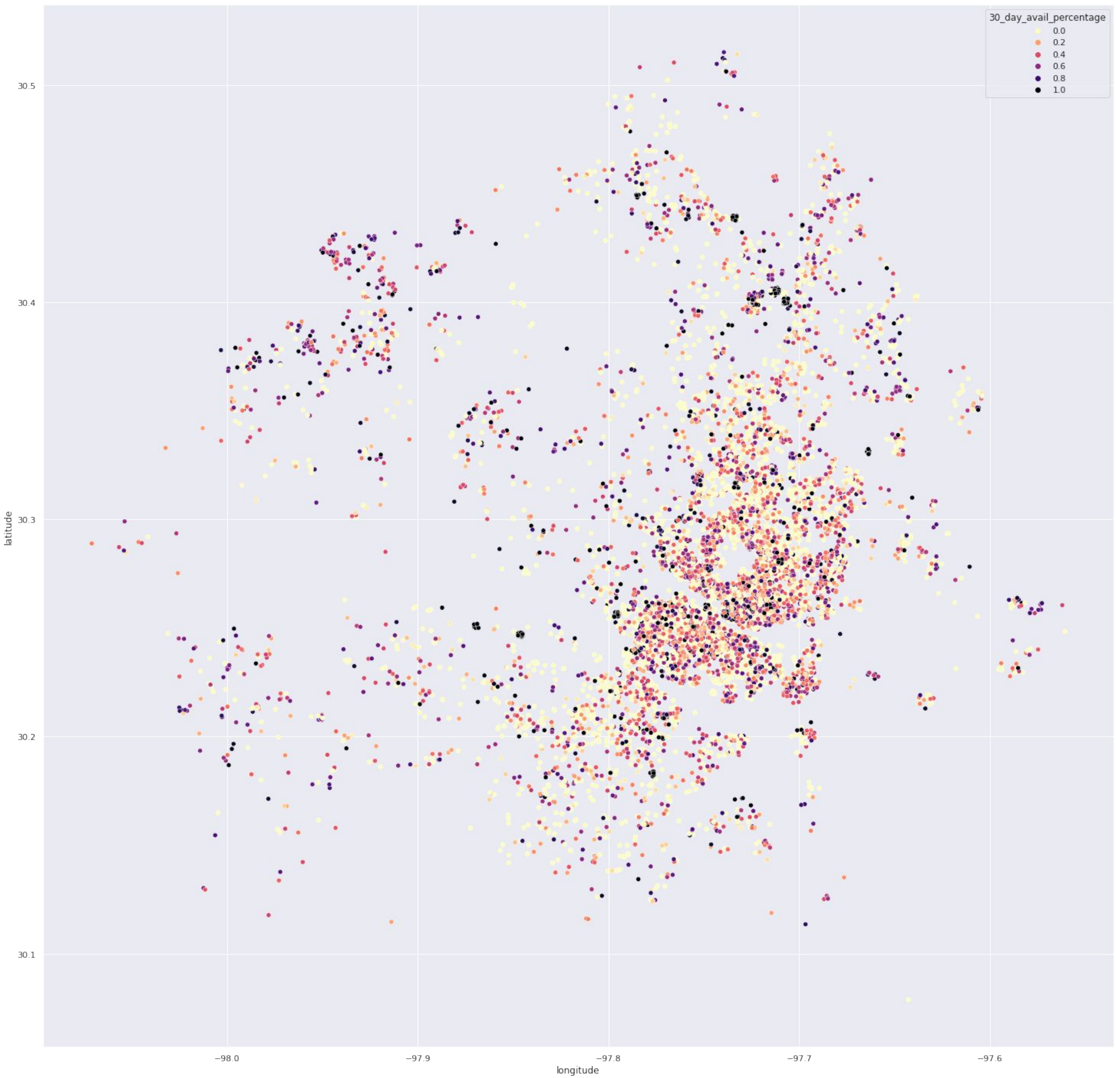
How

- 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)



30_day_avail_percentage
0.0
0.2
0.4
0.6
0.8
1.0
30.5
30.4
30.3
30.2
30.1
latitude
−98.0
−97.9
−97.8
−97.7
−97.6
longitude

Data Insights Cont (Categorization)

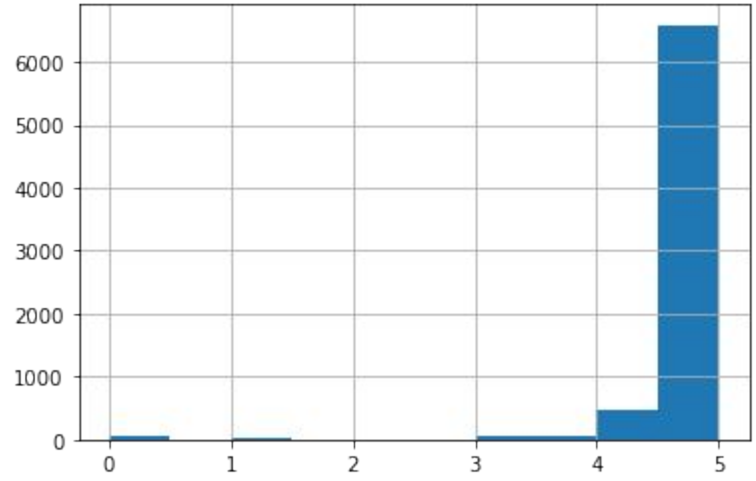
Categorized the following fields:

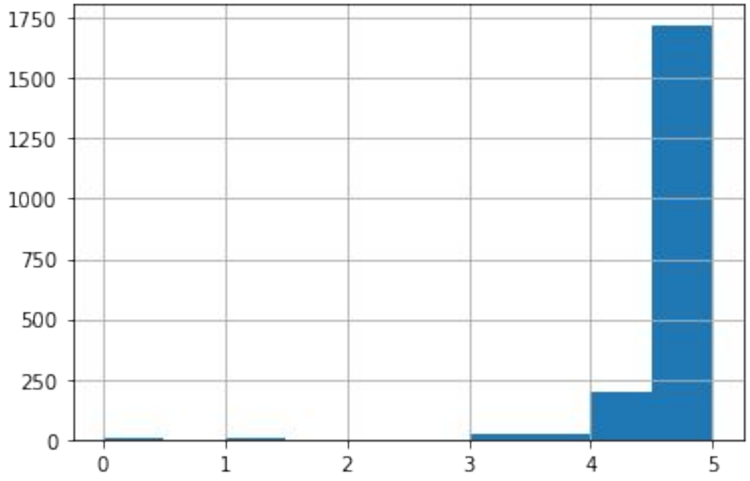
- Neighbourhood
- Property Type
- Top Amenities


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

Data Insights Cont.

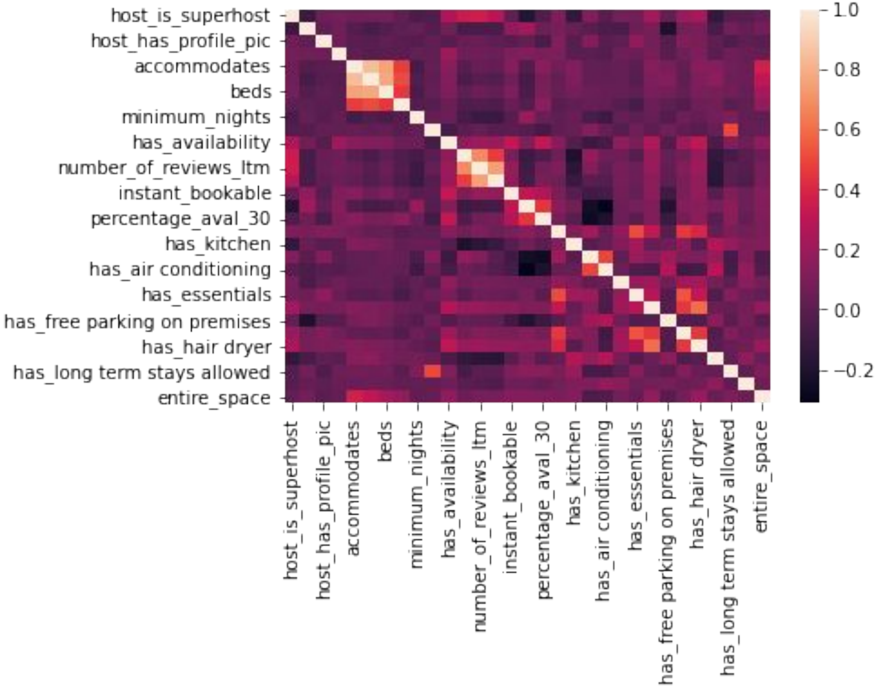
Based on the two groups mentioned earlier many of the attributes in comparison to each other had similar distributions:





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:

- Keras Deep Learning Logistic Regression
- Decision Trees
- Random Forest

Modeling the Data cont. (Optimizing ML Model)

Used a deep learning classification model as a baseline to compare my other models to specifically accuracy, precision and recall. Which raised three questions to be answered:

- Can we reduce number of features?
- Can we improve our models ability to handle variance?
- What metric could we employ to assess model’s effectiveness

Modeling the Data cont. (Streamlining ML model building)

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

```
rfc_pipe = make_pipeline(
    std_scl,
    pca,
    model_rfc
)
```

Model the Data cont. (Optimizing Hyperparameters)

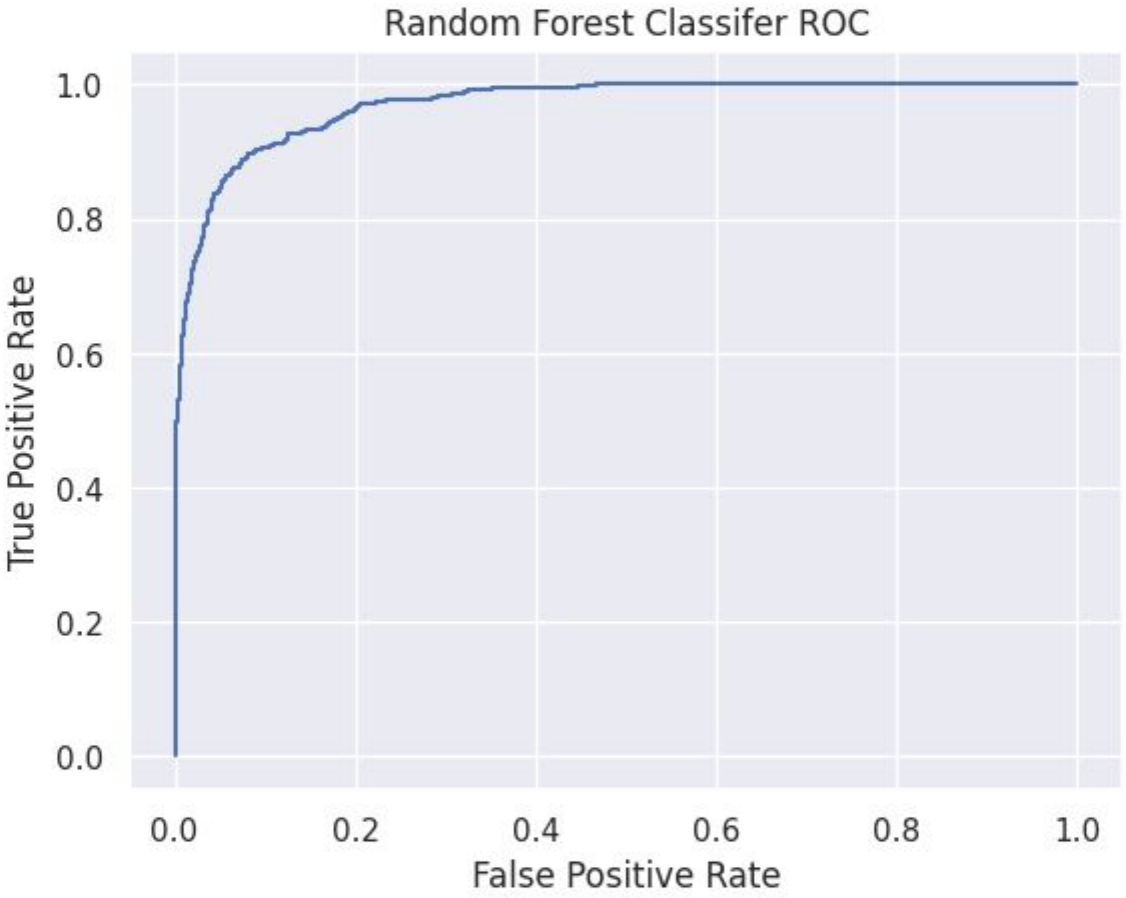
- GridSearchCV
  - Pruning of tree branches for our Decision Tree and Random Forest (Max Depth)
  - Selection of optimal features to use for training the model (PCA)

```
#hyper param tuning/testing
grid_params = {'pca__n_components': list(range(1, X_train.shape[1] + 1, 1)), "randomforestclassifier__max_depth":[4,6,8,10
]}
rfc_grid_cv = GridSearchCV(rfc_pipe, param_grid=grid_params, cv=5, n_jobs=-1)
rfc_grid_cv.fit(X_train, y_train)
```

Model Evaluation

Metrics In respect to our predictions being classified ‘1’

Model Name	Precision (train)	Recall	F1	Accuracy	# of Features
Keras Classifier	0	0	0	0.83	72
Decision Tree Classifier	0.77	0.76	.76	0.92	70
Random Forest Classifier	**0.89**	**0.72**	**0.80**	**0.93**	**13**


Random Forest Classifer ROC
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

Conclusion

- Random Forest Classifier model was selected for:
    - Highest F1 Score
    - Highest Accuracy
    - Least complex model (13 Features)