Data X

About Me:

2Homez

Trent Dill, Vivek Hebbar, Eric Nelson, Aaron Prohofsky, Milad Ravazi

Find Your Vacation Home

- Vacation home sales have increased more than 50% since 2013 [1]
- More than ⅔ vacation homes primary use is recreation or investment [2]
- Our project can recommend vacation home seekers the ideal locations and properties to acquire, based on their specific preferences
    - States or regions preferred
    - Urban vs Rural (population density)
    - Proximity to coast, mountains, national parks
    - Climate and seasonal weather conditions
    - Season(s) desired to live in vs rent out

1 https://www.nar.realtor/news-releases/2015/04/vacation-home-sales-soar-to-record-high-in-2014-investment-purchases-fall
2 https://www.huduser.gov/periodicals/ushmc/spring2004/article_ushmc-04q1.pdf

Berkeley
UNIVERSITY OF CALIFORNIA

Personalize Your Real Estate Investment

- The value of a property = rental income / risk
- Investors use *long-term* rental income to value properties
- Vacation homes produce *short-term* rent which includes a premium
- AirBnB has stores of short-term rental data
- Our project can value short-term rental income that certain properties can expect to make during certain parts of the year
    - Summer income vs Winter income
    - Weekend warrior income vs 6-month tenant income
- **Users can adjust their criteria and choose which seasons they prefer to live in or rent out their property to meet their recreation and investment objectives**
    - **Display mortgage financing options based on their budget constraints**
    - **Project ROI and monthly cash flows for their investment!**

Technical Requirements

High Confidence:

1. Explore data from Zillow and Airbnb in order to have enough overlap for one city to create viable dataset.
2. ETL from Zillow API into cleaned database from properties for sale on Zillow in one given area
3. For each property, extrapolate potential Airbnb seasonal profits from Airbnb data, put into database cross-indexed with sale property database (step 2)
4. Given user input of percents per season, output list of for sale properties sorted by amount of profit made
5. Weight projected season profit by percent of time they plan to rent it out

Low Confidence

1. Creating a model flexible enough to give recommendations outside of user input.
2. Create a model to take a user's budget and suggest locations and usage behaviors.

User Requirements

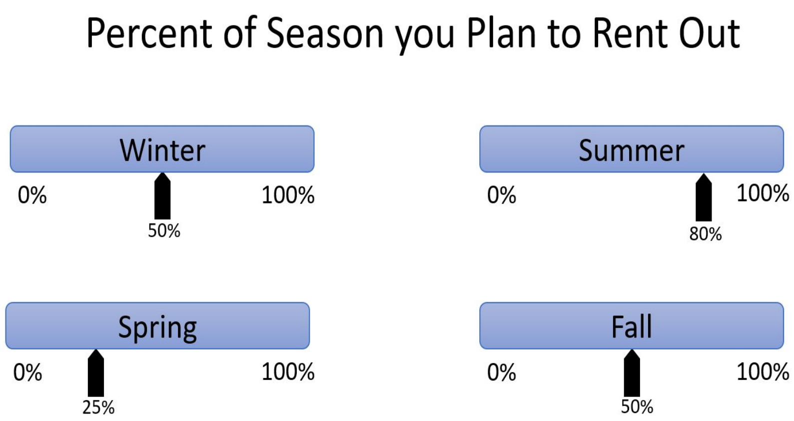
High Confidence:

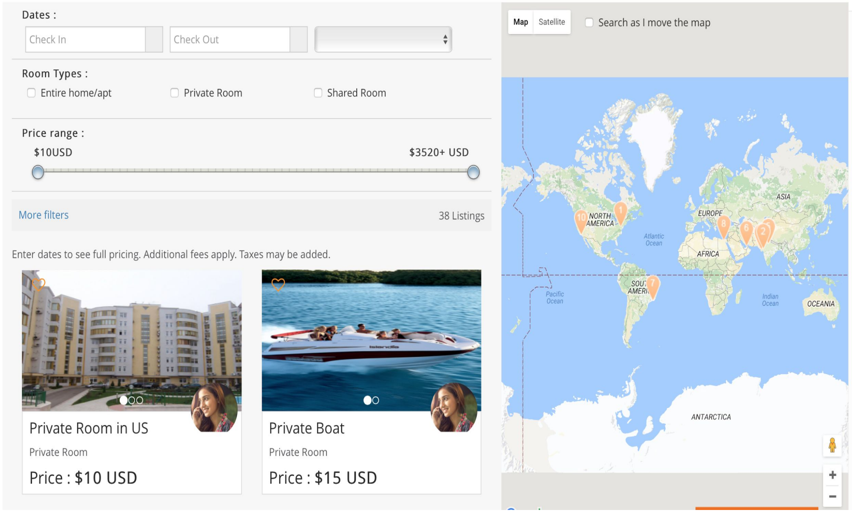
- Be able to see rental income of potential real estate purchase
  - Short term vs Long term
- See available properties for sale in a certain location
- See estimated mortgage for properties in that location vs the estimated short term (Airbnb) rental income
- Percent-of-season slider

Low Confidence:

- Expand to multiple cities
- Visualize, filter properties on a map
- Recommended behavior suggestions- “If you only spend 40% of time here instead of 50%, you could make this much extra”
- Featured Listings - show most profitable
- Comparison of Rental income possibilities

User Interface





2HOMEZ DATA MODEL, ALGORITHM PSEUDO CODE

Rough Draft I

Step 1: Populate Database 'for_sale' with properties for sale in city on Zillow. Use Zillow API.

for_sale

Address	Lat	Lon	Bed	Bath	Nbrhd	Mortgage
. . .						

Step 2: Populate Database 'air_data' with properties in city found in Airbnb archive data. Use insideairbnb.

air_data

Address	Lat	Lon	Bed	Bath	Nbr hd	WRate	SpRate	SumRate	FallRate

. . .

Step 3: Either:

1. One hot encode Neighborhood → Train model on entire air_data table: $F(Bed, Bath, Nbrhd, \ldots) = \underbrace{WRate, SpRate, SuRate, FaRate}_{\text{what model outputs}}$

   ↓

   apply model to each row in for_sale

OR

2. For each row in for_sale, select training set from air_data consisting of nearby Airbnb houses

   ↓

   train model on small per-row dataset, apply to row: $F(Bed, Bath, \ldots) = WR, SpR, SuR, FaR$

Output: four new columns appended to the 'for_sale' table:

for_sale

Address	Lat	Lon	...	Mortgage	Proj_WRate	Proj_SpRate	Proj_SuRate	Proj_FaRate

Step 4: User input: % of winter they will rent $= pWI$

% of spring they will rent $= pSP$

% of summer they will rent $= pSU$

% of fall they will rent $= pFA$

- We want to find properties – rows in for_sale – with projected rate proj_WRate, proj_SpRate, proj_SuRate, proj_FaRate such that $\underbrace{\boxed{pWI \times proj\_WRate + pSP \times proj\_SPRate + pSU \times proj\_SuRate + pFA \times proj\_FaRate}}_{air\_score}$ is **maximized**.

For every user query (pWI, pSP, pSU, pFA), add column to for_sale. The column to add is air_score(query, property) for every property in for_sale. We then sort for_sale by air_score in descending order and display to the user:

for_sale

Address	Lat	Lon	. . .	projFaRate	air_score ↓

Show top x properties to user

Step 5: Using gradient descent or simulated annealing, find values for pWI, pSU, pSP, pFA that maximize air_score for top x properties. These are suggestions that are provided to the user to help them maximize their short-term rental potential profits.

You inputted: pWI, pSP, pSU, pFA.

If you instead use: pWI', pSP', pSU', pFA', here is how your profits would change:

for_sale

Addr	Lat	Lon	. . .	projFaRat	air_score_updated ↓

]:
```
sf_prices = sf_avail.dropna().drop(['date','available'],axis=1)
sf_prices['price'] = sf_prices['price'].replace( '[\$,)]','', regex=True ).replace( '[(]','-',
sf_prices = pd.pivot_table(sf_prices,index=['listing_id'], columns=['Season'],aggfunc=np.mean)
sf_prices.head(10)
```

]:

	price			
Season	**Fall**	**Spring**	**Summer**	**Winter**
listing_id				
958	171.560000	171.695652	171.783784	171.733333
5193	175.000000	160.000000	167.195122	161.518987
5841	183.432432	183.543478	183.903226	183.600000
5858	250.000000	250.000000	250.000000	250.000000
7918	65.000000	65.000000	65.000000	65.000000
8014	60.000000	60.000000	60.000000	60.000000
8142	65.000000	65.000000	65.000000	65.000000
8339	395.000000	NaN	395.000000	NaN
8739	188.695122	410.108696	333.310811	187.800000
8775	344.924242	285.000000	283.743590	285.000000

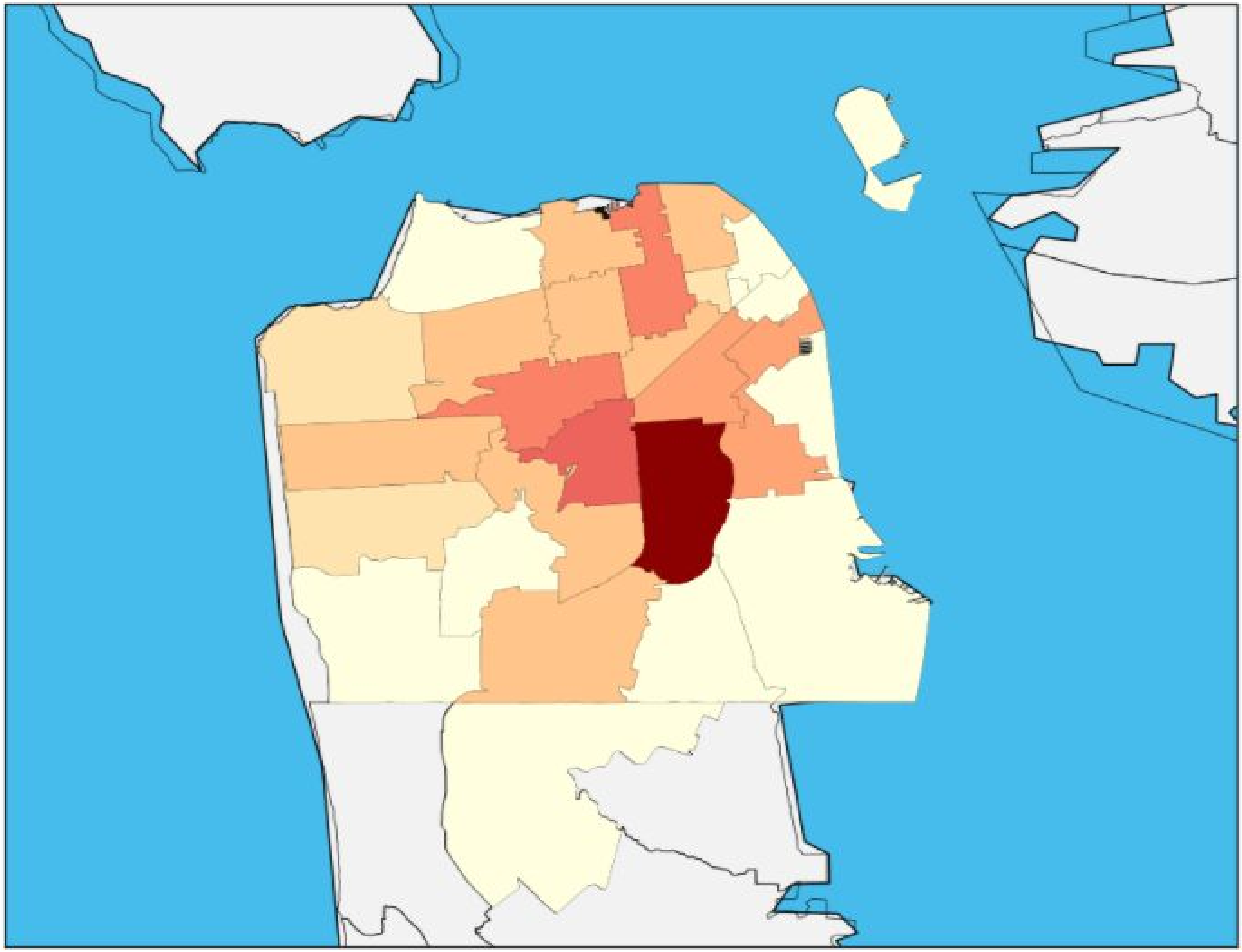
:
```
sf_season_avail = sf_avail.drop(['date','price'],axis=1)

def calc_avail(x):
    vc = x.value_counts()
    if 't' in vc.keys():
        return x.value_counts()['t'] * 1.0 / x.size
    else:
        return 0

sf_season_avail = pd.pivot_table(sf_season_avail, index=['listir
sf_season_avail.head()
```

:

	available			
Season	**Fall**	**Spring**	**Summer**	**Winter**
listing_id				
958	0.549451	1.0	0.402174	1.000000
5193	0.340659	1.0	0.445652	0.877778
5841	0.813187	1.0	0.336957	1.000000
5858	1.000000	1.0	0.869565	1.000000
7918	1.000000	1.0	1.000000	1.000000



Value Proposition

- The US Government incentivizes homeownership with tax deductions
    - Mortgage interest is 100% tax deductible
    - Property tax is 100% tax deductible
    - Capital Gains are exluded from taxation (up to $250K or $500K for married couples)
- These tax benefits apply to up to **two homes per person**
    - Professional investors miss this value, our users can capitalize
    - Most two-homeowners are in the top tax bracket -> more savings
- Our project connects the resources of a real estate investment firm with the tax-savings potential of a single homeowner to optimize vacation home investment
- By changing the owner from a real estate investor to a single-homeowner, our project creates scalable tax savings out of thin air
    - For arranging the acquisition, we could charge a portion of these savings

Monetization

Customer acquisition for mortgage bankers: mortgage bankers are hungry to dish out interest-earning mortgages and for referring customers to bankers, and we could charge a fee.

Loan Discount Fee: we could give the borrower less money than they are required to repay. This “loan discount fee” is common in real estate and can be rationalized by the tax savings. It could be ½ of the NPV of the tax savings the user will realize, meaning the user still pockets half. We can partner with mortgage lenders to be able to offer this type of financing, and the partnership makes sense for the mortgage lender because we are acquiring customers on their behalf.

Next Steps: Week 7

Understand	Code	Validate
• How to extrapolate data on number of bedrooms and bathrooms	• Incorporate Zillow data into model to start predicting on	• Run regression models and compare their accuracy and find ways to optimize the structure of our model