# Predicting Number of Days on the Market



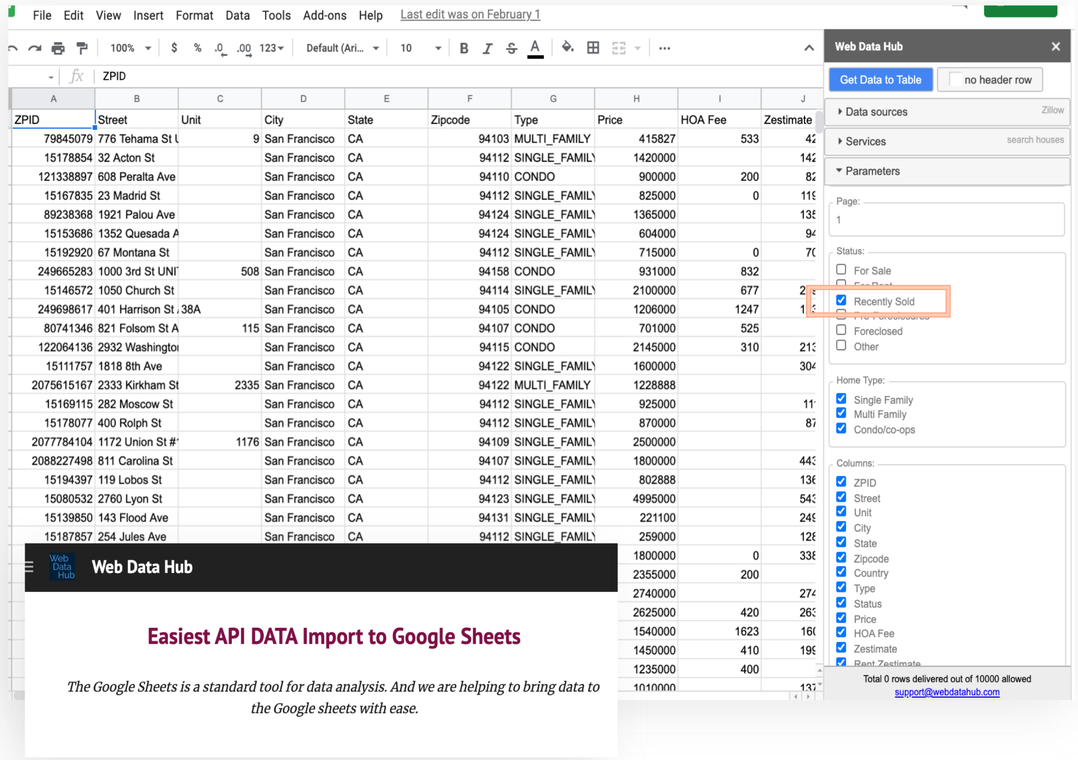
Ariel Chen

# Framing the Problem

The **# of days on the market** is a metric that many in the real estate industry care about whether it be sellers who want to ensure that their house isn't sitting on the market for too long or buyers who are curious about the neighborhood desirability.

This project aims to create a model that predicts the number of days a house in the Bay Area is projected to be on the market for.

# Process Approach




## 1. Acquiring the Necessary Dataset

From my research, I knew I need a dataset that would include individual listings.

- X Redfin
- X Zillow
- X Realtor.com

☺ Web Data Hub

### Limitations:

- Allows only 200 listings to be pulled at a time
- Counter at bottom limiting max number of what you could pull

### Workaround:

- Had a choice of Zillow or Realtor.com, Zillow option had more filters (including the main one: Days on Zillow)
- Pulled 200 at a time, copy pasting into Excel to compile into master dataset

# Process Approach

**ZPID** Used to filter out duplicate entries

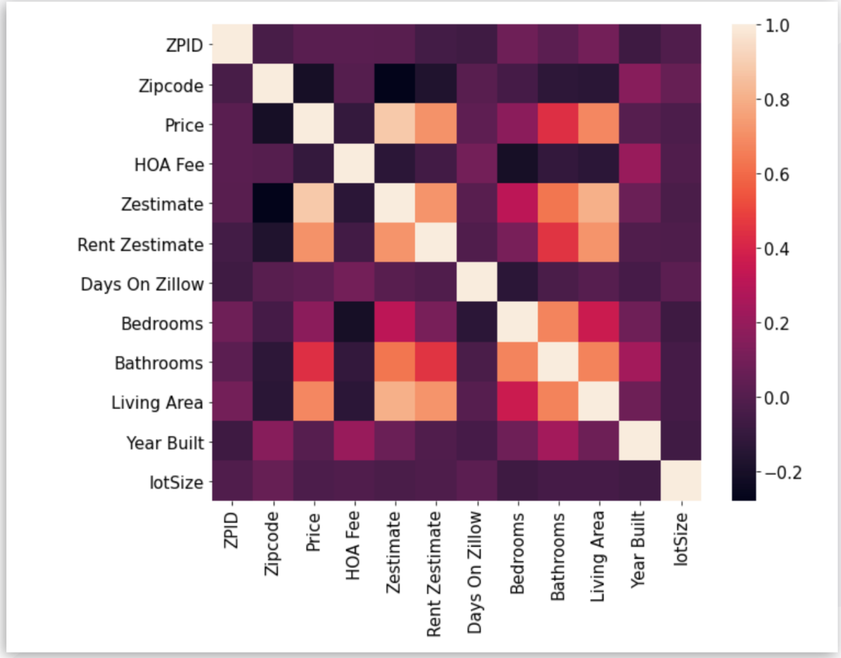
**Days On Zillow** Take out rows with no data in this column as this is target variable; removed outliers

**Living Area**

**Price**

**Lot Size**

Take out rows with no data here**




I conducted exploratory analysis of the data which contained **4001** rows pulled from Zillow.

***Cut down to 507 listings***

## 2. Parsing / Refining Data

** Note: these variables were ones that I initially thought would be included in the model but didn't later

# Process Approach

**Zipcode**

Grouped by first 3 digits; converted to be interpreted as a categorical variable→ pd.get_dummies

```
# stats f_oneway functions takes the groups as input and returns ANOVA F a
fvalue, pvalue = stats.f_oneway(zipcode_dummies['Zipcode_940'], zipcode_du
print('F statistic = {:5.3f} and probability p = {:5.3f}'.format(fvalue, p
```

```
F statistic = 364.475 and probability p = 0.000
```

Ran ANOVA test to test null hypothesis of no relationship between Zipcode + other variables

**P-value = 0.000 → reject null hypothesis**

| zipcode_alpha | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| 940 | 110.0 | 34.581818 | 35.278026 | 0.0 | 6.25 | 18.0 | 55.75 | 140.0 |
| 941 | 71.0 | 28.887324 | 39.966968 | 0.0 | 4.50 | 11.0 | 25.00 | 148.0 |
| 943 | 19.0 | 55.526316 | 46.645672 | 1.0 | 10.00 | 47.0 | 90.00 | 149.0 |
| 944 | 26.0 | 38.807692 | 42.358488 | 0.0 | 7.00 | 18.0 | 66.75 | 131.0 |
| 945 | 65.0 | 20.415385 | 27.397475 | 0.0 | 3.00 | 7.0 | 27.00 | 118.0 |
| 946 | 1.0 | 124.000000 | NaN | 124.0 | 124.00 | 124.0 | 124.00 | 124.0 |
| 947 | 8.0 | 38.375000 | 37.640926 | 4.0 | 9.25 | 32.0 | 55.50 | 110.0 |
| 949 | 7.0 | 32.857143 | 47.928518 | 6.0 | 11.00 | 16.0 | 23.00 | 140.0 |
| 950 | 48.0 | 33.125000 | 32.438239 | 0.0 | 10.75 | 23.0 | 52.25 | 146.0 |
| 951 | 124.0 | 35.274194 | 37.506712 | 0.0 | 6.00 | 22.0 | 52.25 | 145.0 |

Although P-value from the ANOVA test was 0.000, analyzed zipcode more closely and it seems that zipcode isn't very significant as the ranges for "Days on Zillow" are somewhat similar across the groupings.

**Business decision = include it in the model as it is a big consideration factor when searching for homes**

## 3. Building the Model

# Performance Evaluation

```python
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn import metrics
from sklearn.ensemble import RandomForestRegressor

# Define a function that accepts a list of features and returns testing RMSE.
def train_test_cv_lr(df, feature_cols):
    X = df[feature_cols]
    y = df['Days On Zillow']

    lr = LinearRegression()
    rf = RandomForestRegressor()

    return np.mean(np.sqrt(-cross_val_score(lr, X, y, scoring='neg_mean_squared_error', cv=3)))

def train_test_cv_rf(df, feature_cols):
    X = df[feature_cols]
    y = df['Days On Zillow']

    lr = LinearRegression()
    rf = RandomForestRegressor()

    return np.mean(np.sqrt(-cross_val_score(rf, X, y, scoring='neg_mean_squared_error', cv=3)))
```

## Considered linear regression model + random forest model for prediction

```
Linear Regression MSE
36.54485910988583
38.30273541581384

Random Forest MSE
43.107426347029985
44.936003819990695
```

Tested out same variables for both models to find RMSE.

** The lower the error the better*

**Predictive model → linear regression**

## 3. Building the Model

# Performance Evaluation / Finding Impacts

```
print(train_test_cv_lr(zipcode_dummies_all, ['Zipcode_940','Zipcode_941', 'Zipcode_943', 'Zipco
print(train_test_cv_lr(zipcode_dummies_all, ['Living Area', 'HOA Fee', 'Year Built', 'Zipcode_9
print(train_test_cv_lr(zipcode_dummies_all, ['Bathrooms', 'HOA Fee', 'Year Built', 'Zipcode_941
print(train_test_cv_lr(zipcode_dummies_all, ['Bathrooms', 'HOA Fee', 'Year Built', 'Zipcode_941
print(train_test_cv_lr(zipcode_dummies_all, ['Bedrooms', 'Price', 'Year Built', 'Zipcode_941','
print(train_test_cv_lr(zipcode_dummies_all, ['Price', 'HOA Fee', 'Year Built', 'Zipcode_941','Z
print(train_test_cv_rf(zipcode_dummies_all, ['lotSize', 'HOA Fee', 'Year Built', 'Zipcode_941',
print(train_test_cv_rf(zipcode_dummies_all, ['Living Area', 'HOA Fee', 'Year Built', 'Zipcode_9
```

```
39.0362864975741
38.68706694217594
38.78754170005258
38.78754170005258
38.436434433276595
38.4099972170735
42.45194407515441
41.854679691395255
```

```
Linear Regression MSE
36.54485910988583
38.30273541581384

Random Forest MSE
43.107426347029985
44.936003819990695
```

**Tested to see if other variable combos would give better performance RMSE**

Bedrooms, Year Built, HOA Fee, and Zipcode had the best performance →**RMSE 36.544**

Different from what I was going to use in the model initially (Zipcode, Price, Living Area; RMSE 39.03)

## 3. Building the Model

# Performance Evaluation / Finding Impacts

Utilized variables in prior slide to build linear regression model

Defined function to automate / make using the predictive model easier to use

```
In [42]: feature_cols = ['Bedrooms', 'HOA Fee', 'Year Built', 'Zipcode_941','Zipcode_943', 'Zipcode_944'
         X = zipcode_dummies_all[feature_cols]
         y = zipcode_dummies_all['Days On Zillow']

         lr = LinearRegression()
         lr.fit(X, y)

Out[42]: LinearRegression()

In [43]: ##have to edit auto predict function to match new variable inputs
         def autopredict(Bedrooms, HOA, Year, Zipcode):
             zippy = Zipcode //100
             if zippy == 940:
                 return lr.predict([[Bedrooms, HOA, Year,0,0,0,0,0,0,0,0]])
             if zippy == 941:
                 return lr.predict([[Bedrooms, HOA, Year,1,0,0,0,0,0,0,0]])
             if zippy == 943:
                 return lr.predict([[Bedrooms, HOA, Year,0,1,0,0,0,0,0,0]])
             if zippy == 944:
                 return lr.predict([[Bedrooms, HOA, Year,0,0,1,0,0,0,0,0]])
             if zippy == 945:
                 return lr.predict([[Bedrooms, HOA, Year,0,0,0,1,0,0,0,0]])
             if zippy == 946:
                 return lr.predict([[Bedrooms, HOA, Year,0,0,0,0,1,0,0,0]])
             if zippy == 947:
                 return lr.predict([[Bedrooms, HOA, Year,0,0,0,0,0,1,0,0]])
             if zippy == 949:
                 return lr.predict([[Bedrooms, HOA, Year,0,0,0,0,0,0,1,0]])
             if zippy == 950:
                 return lr.predict([[Bedrooms, HOA, Year,0,0,0,0,0,0,0,1,0]])
             if zippy == 951:
                 return lr.predict([[Bedrooms, HOA, Year,0,0,0,0,0,0,0,0,1]])
```

## 3. Building the Model

Using the model

# Example

**Finding the predicted Number of Days on Zillow for a house with:**

- 3 bedrooms
- No HOA Fee
- Built in 1978
- Zipcode 94130

```
autopredict(# bedrooms, HOA fee, Year built, Zipcode)
```

```
autopredict(3, 0, 1978, 94130)
```

```
array([25.98775745])
```

**The predicted number of days on Zillow for this property would be ~26 days**

# Next Steps

- Find a way to get access to an automatically generated dataset that applies not just to the Bay Area. That way, the model could be more applicable to more people.

- Deploy and host the model online so that people could use the model without having to open jupyter notebook.

***This predictive model could help folks predict the number of days a house will be on the market for whether it be for selling / buying purposes.***

# THANK YOU.