A Neighborhood's Effect on its Housing Market

Sonjá Williams
Olivia Blocker
Jacob Odetunde
Dan Beres

Agenda

- Reason for the project
- Statement of problem
- Tools, Technology and algorithms
- Data sources
- Data exploration
- Data Analysis
  - Machine Learning Code
  - Visualizations
- Challenges
- Next Steps/Future Research

How does crime and demographic data affect housing value in Illinois?

Our goal with these data sets was to try and find any correlation of factors that might affect housing value. Using Zillow data we are going off of Zillow Home Value Index (ZHVI) as our way of assigning value to properties.

Factors accounted for:

- Cities in Illinois
- Demographic information
- Crime Rates of the Cities

Potential Questions:

- What factors most highly impact ZHVI?
- How much do these uncontrollable factors influence ZHVI?

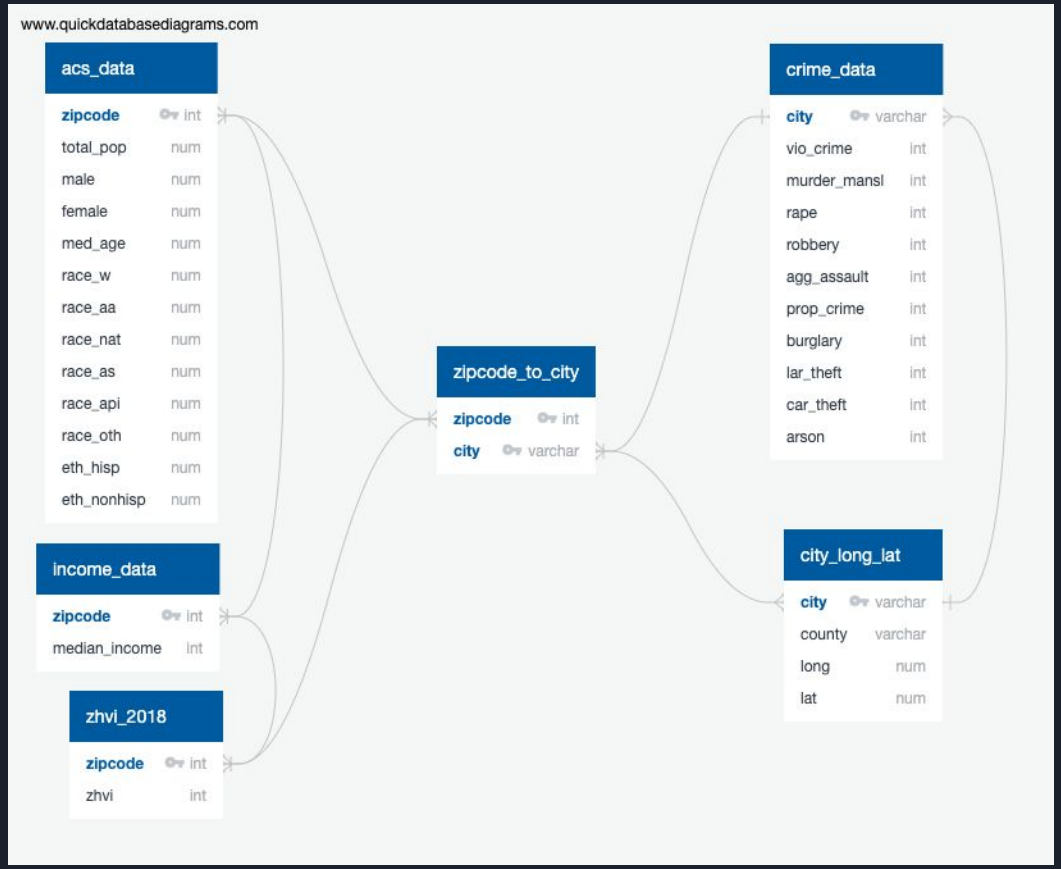
Technologies

- AWS database
- PostgresSQL
- Machine Learning
  - PySpark taking information from tables in PostgreSQL
  - Supervised learning with scikit-learn library
- Tableau
  - Important figures imported through iframes into a Dashboard+
- GitHub Pages
  - Dashboard was created through github pages, using HTML5/js to help show and filter information.

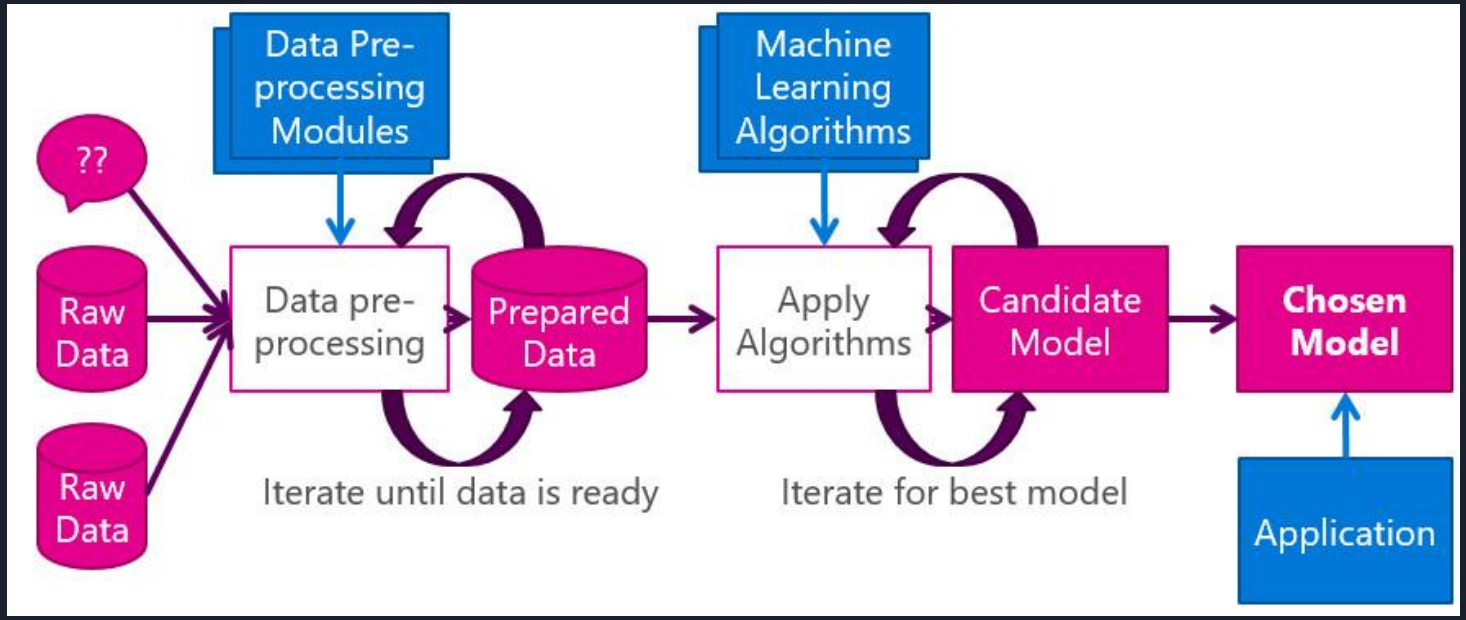
Sources of Data

- Zillow
  - Market dataset broken down by city and zip code
- US Census
  - Primarily used for demographic and income information
- Department of Justice
  - FBI Uniform Crime Reporting gives state-wide reported crimes by city
- OpenDataSoft
  - Offers latitudinal and longitudinal data for cities to better help with visualizations

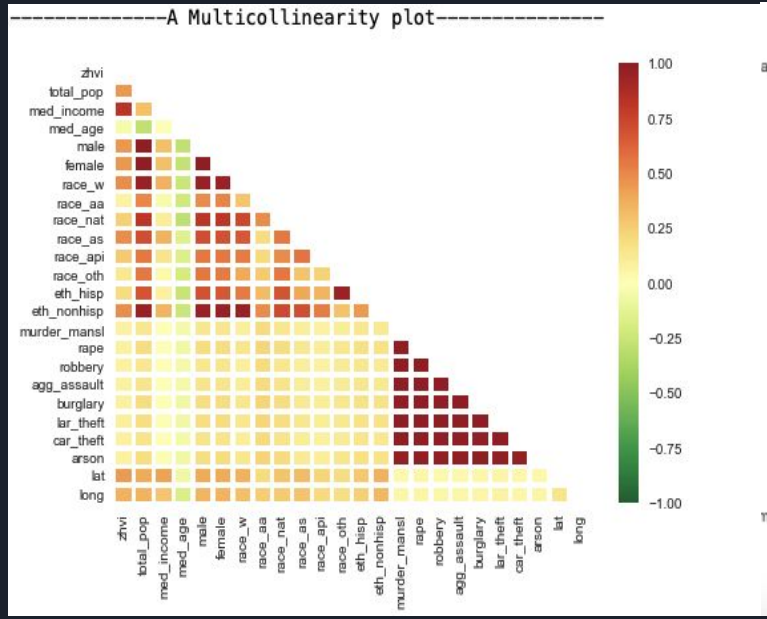
Data Exploration

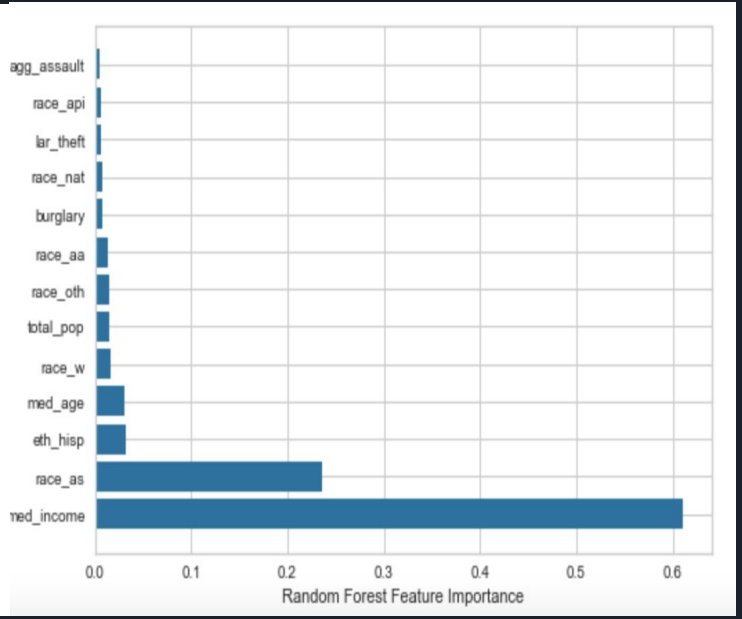


Overview of Model's Processes



Feature Engineering





Algorithms

```
model_1 = LinearRegression()
model_1.fit(X_train,y_train)
predictions = model_1.predict(X_test)

print(" MAE", mean_absolute_error(y_test,predictions))
print(" RMSE", sqrt(mean_squared_error(y_test,predictions)))
print(" R2", r2_score(y_test, predictions))
```

```
MAE 33203.988282283026
RMSE 52409.26262287243
R2 0.7508905264445278
```

```
model_2 = KNeighborsRegressor()
model_2.fit(X_train,y_train)
predictions = model_2.predict(X_test)

print(" MAE", mean_absolute_error(y_test,predictions))
print(" RMSE", sqrt(mean_squared_error(y_test,predictions)))
print(" R2", r2_score(y_test, predictions))
```

```
MAE 32951.56553571428
RMSE 56251.162487599024
R2 0.713029569544966
```

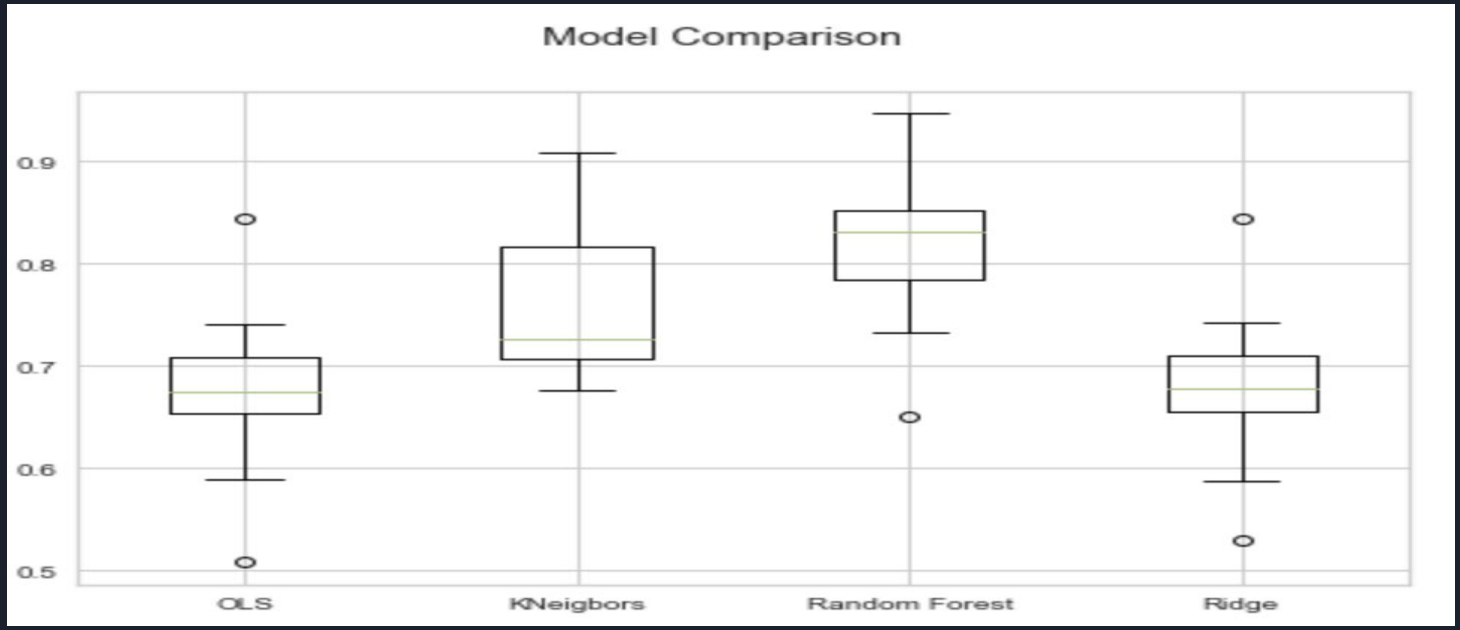
```
model_3 = RandomForestRegressor()
model_3.fit(X_train,y_train)
predictions = model_3.predict(X_test)

print(" MAE", mean_absolute_error(y_test,predictions))
print(" RMSE", sqrt(mean_squared_error(y_test,predictions)))
print(" R2", r2_score(y_test, predictions))
```

```
MAE 26380.365680625582
RMSE 45308.48824113064
R2 0.8138198873591888
```

```
model_4 = Ridge()
model_4.fit(X_train,y_train)
predictions = model_4.predict(X_test)

print(" MAE", mean_absolute_error(y_test,predictions))
print(" RMSE", sqrt(mean_squared_error(y_test,predictions)))
print(" R2", r2_score(y_test, predictions))
```

```
MAE 33100.91334924512
RMSE 52349.41955861106
R2 0.7514590886605582
```

Model Selection



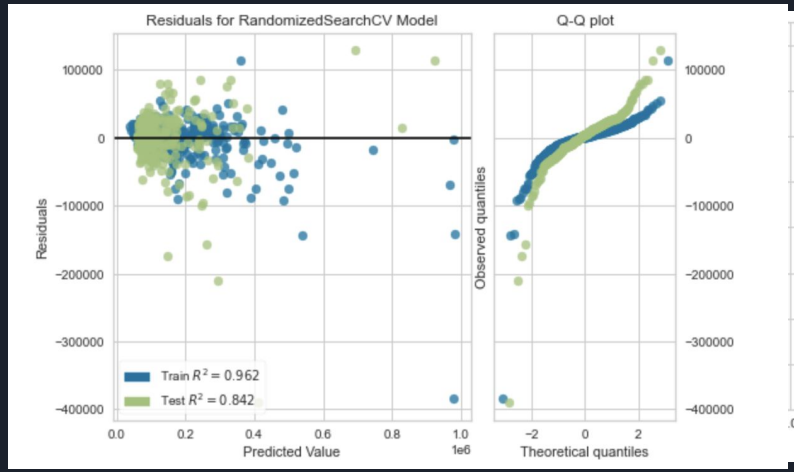
Model Parameters

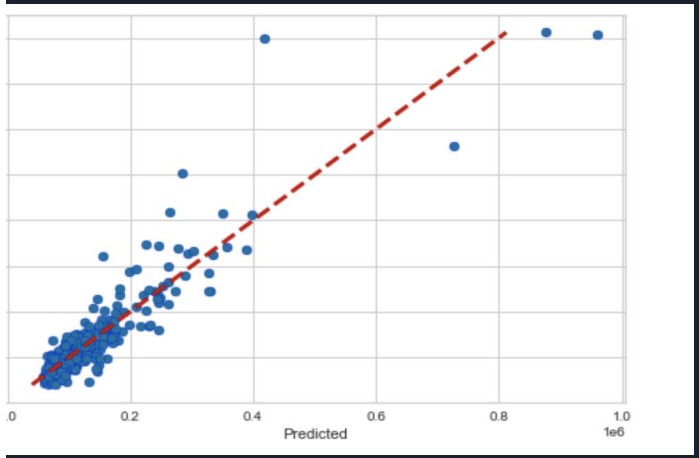
```
# Search for the best params
param_dist = {"max_depth": [3, None],
              "max_features": sp_randint(1, X_train.shape[1]),
              "min_samples_split": sp_randint(2, 11),
              "bootstrap": [True, False],
              "n_estimators": sp_randint(100, 500)}

random_search = RandomizedSearchCV(model, param_distributions=param_dist,
                                   n_iter=10, cv=5, iid=False, random_state=42)
best_model = random_search.fit(X_train, y_train)
print(random_search.best_params_)

y_preds = best_model.predict(X_test)
print(r2_score(y_test, y_preds))
```

```
/Users/jacob/opt/anaconda3/envs/PythonData/lib/python3.8/site-packages/sklearn/model_selection/_search.py:847: Futu
reWarning: The parameter 'iid' is deprecated in 0.22 and will be removed in 0.24.
  warnings.warn(
```

```
{'bootstrap': True, 'max_depth': None, 'max_features': 11, 'min_samples_split': 9, 'n_estimators': 288}
0.8360061308203486
```

Model Evaluation





Mean Absolute Error: 24972.866154346626
Root Mean Squared Error: 41779.7211562642

Predictions

	city	Prediction	Actual
0	Ladd	92093.144459	83739.500000
1	Lake Bluff	139689.897010	131423.580000
2	Lake Forest	163870.008611	162155.670000
3	Lake Villa	76990.890586	91324.080000
4	Lake Zurich	69512.074465	81336.580000
5	Lakewood	103817.233722	136188.750000
6	Lanark	88465.069851	88259.580000
7	Lansing	242275.921895	229937.303333
8	Latham	78591.441775	52089.750000
9	Lawndale	81377.928232	111396.250000

Pros and Cons

We decided to choose Random Forest Regressor because of the following reasons:

- It gives the highest R-square value compare to other algorithms.
- It gives minimum Mean Absolute Error(MAE) and Root Mean Square Error (RMSE) compare to other algorithms we evaluated.
- It reduces overfitting in decision trees and helps to improve the accuracy
- Random Forest can automatically handle missing values.
- Random Forest is usually robust to outliers and can handle them automatically.

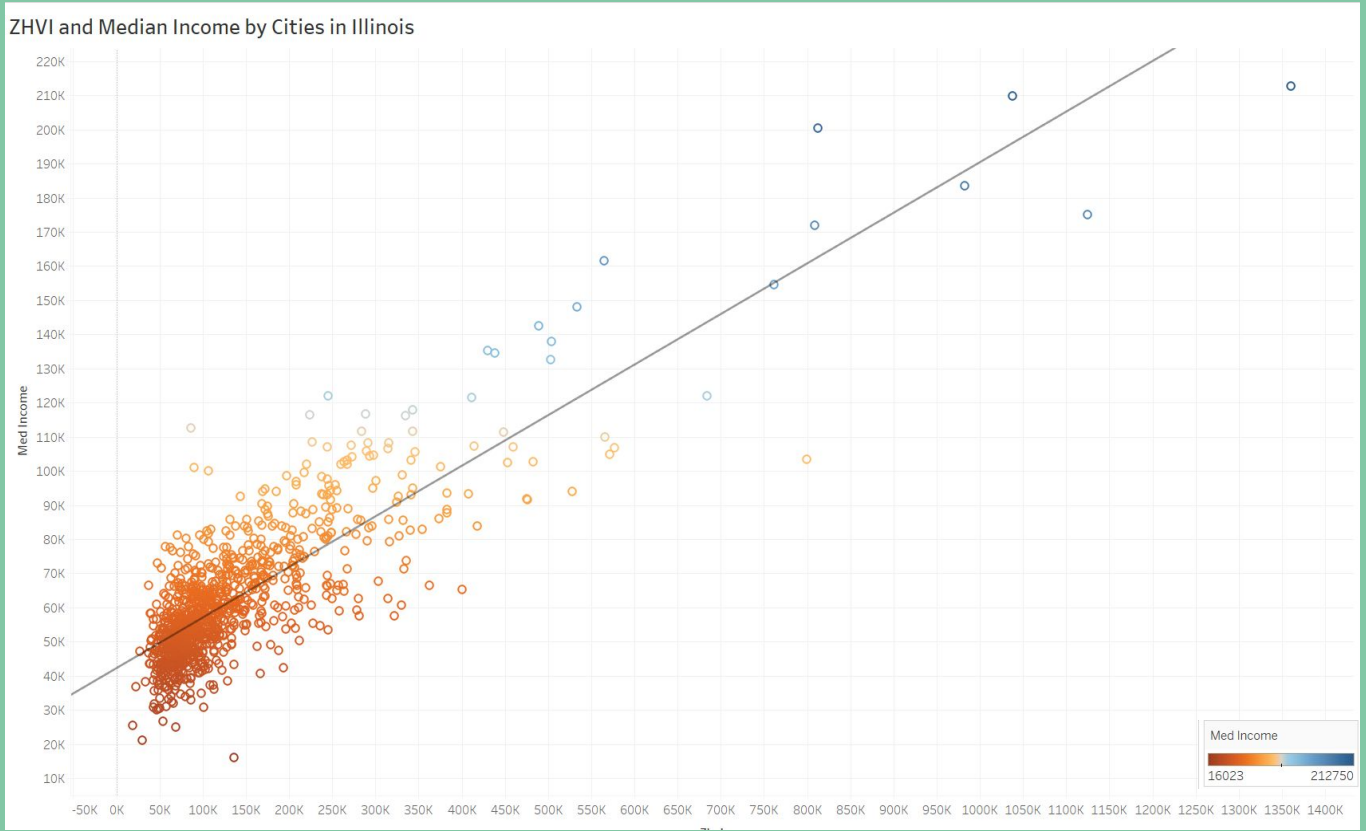
Limitation of model selected

- Random Forest require much more time to train as compared to decision trees as it generates a lot of trees (instead of one tree in case of decision tree) and makes decision on the majority of votes. -It requires much computational power as well as resources as it builds numerous trees to combine their outputs.
- Due to the ensemble of decision trees, it also suffers interpretability and fails to determine the significance of each variable.

Main Stats

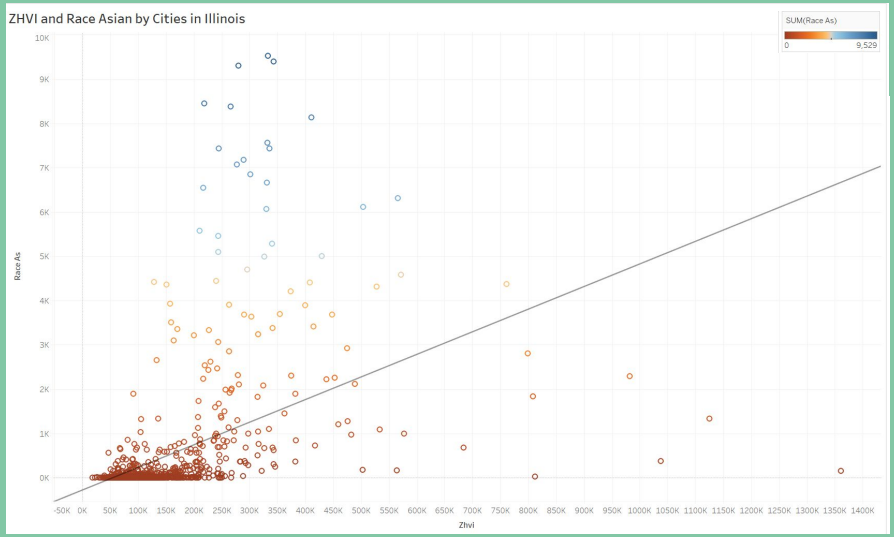
Top 3 Factors

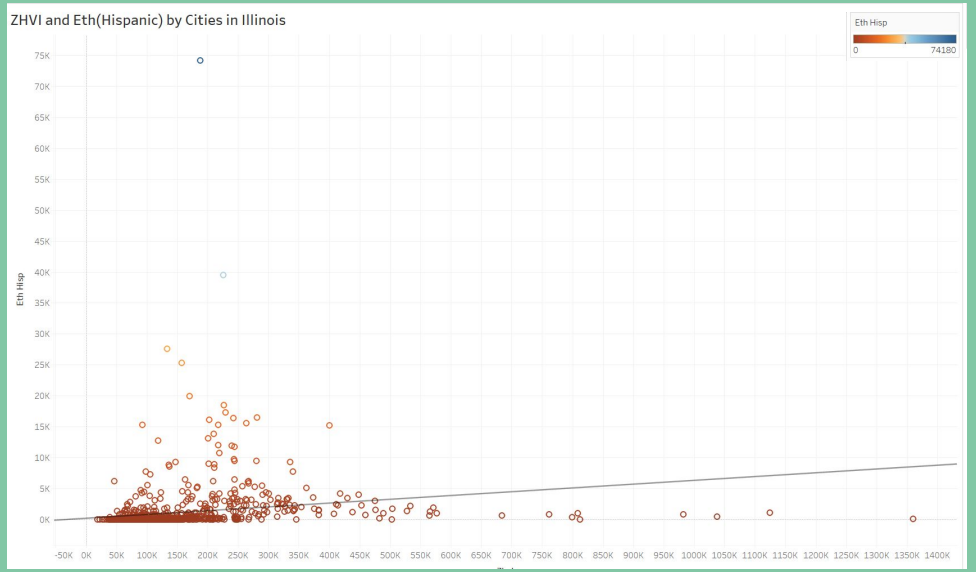
- Median Income
  - Shows the greatest Positive Correlation



Main Stats

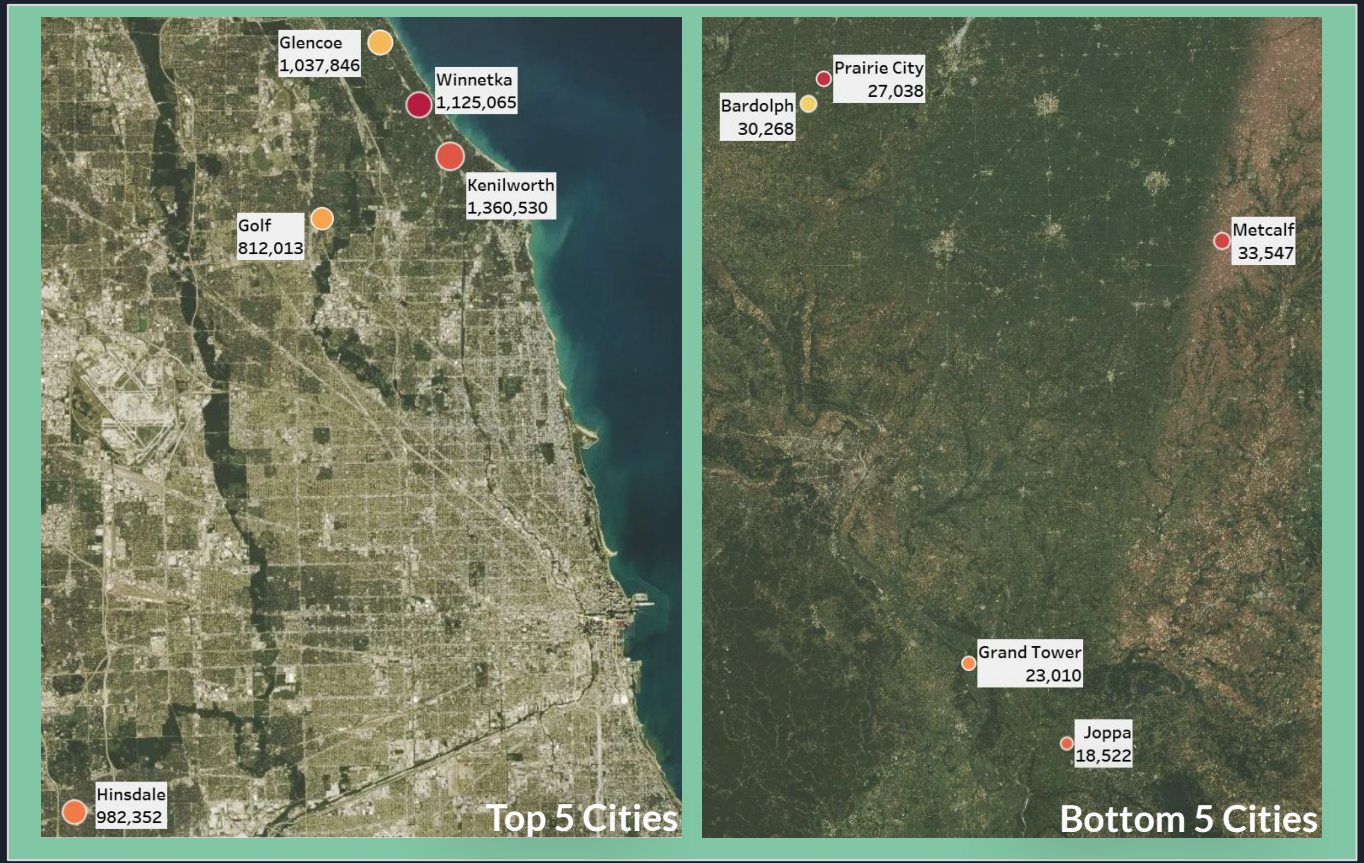
- (Race) Asian
  - 5.5% of total population
  - 8.5% of Top 5 Cities Ranked by ZHVI





- Ethnicity Hispanic
  - 13.8% of total population
  - 5.1% of Top 5 Cities Ranked ZHVI
  - Cicero significant outlier (pop.74,180)

Main Stats


Glencoe
1,037,846
Winnetka
1,125,065
Kenilworth
1,360,530
Golf
812,013
Hinsdale
982,352
Top 5 Cities
Prairie City
27,038
Bardolph
30,268
Metcalf
33,547
Grand Tower
23,010
Joppa
18,522
Bottom 5 Cities
City
Bardolph
Glencoe
Golf
Grand Tower
Hinsdale
Joppa
Kenilworth
Metcalf
Prairie City
Winnetka
SUM(Zhvi)
18,522
500,000
1,000,000
1,360,530

The Find Your City Tool

755	Wayne	503301.33	2357.0	50.4	39.0	2318.0	4.0	41.950861	-88.242297
767	Western Springs	564362.92	13227.0	41.7	633.0	12594.0	139.0	41.809753	-87.900616
781	Willowbrook	527717.67	27903.0	46.6	1371.0	26532.0	0.0	41.457813	-87.536429

In [18]:
```
# Add a heatmap of temperature for the vacation spots.
locations = city_info[["lat", "long"]]
price = city_info["zhvi"]
fig = gmaps.figure(center=(30.0, 31.0), zoom_level=1.5)
heat_layer = gmaps.heatmap_layer(locations, dissipating=False,
                 max_intensity=300, point_radius=4)
marker_layer = gmaps.marker_layer(locations)
fig.add_layer(heat_layer)
fig.add_layer(marker_layer)
# Call the figure to plot the data.
fig
```

In [19]:
```
info_box_template ="""
<dl>
<dt>city</dt><dd>{city}</dd>
<dt>zhvi</dt><dd>{zhvi}</dd>
```

Challenges

- Finding functional datasets
- Merging / cleaning the data
- Machine learning model selection
- Introducing our biases
- Finding a story & choosing visuals

Next Steps

- Future Research / Features
  - Exploratory Data Research: our feature_importances_ algorithm listed the Asian race and Hispanic ethnicity as a particularly important demographic feature that impacts housing value.
  - Trend Analysis: How can we use this factors to help predict which cities will have better home values in the future?
  - Develop a Chicago-specific model for investors
    - There is a lot more data (specifically crime data) available for Chicago
    - Chicago is an outlier
  - Develop a user-friendly interface for investors to find their dream location
    - Adding data fields such as city type (rural, suburban, and urban), and cost of living data