Real Estate Trends in Boston

Joe Driver **Alex Lammers** **Justin Lee** **Xiaotong Niu**

Abstract

Over the last twenty years the the cost of living in the city of Boston has changed dramatically. Areas such as South Boston and the Seaport have gone from dangerous no-go zones to some of the most sought after properties in the area.

Although incomplete in many years, the city and other groups have done an admiral job of recording these changes in a semi-consistent group of datasets, found at `https://data.boston.gov/dataset/property-assessment`. For data provided over 14 years (2004 - 2017) our goal is to gather property information regarding condition, size, type, and value; and use machine learning techniques to project the change in that value of over time. With this information we can identify "hot-spots" of growing value, areas of rapid value increase that may push lower income residents out of the city, and possibly where the next best place to buy will be.

For this analysis we are focusing on residential properties in the city of Boston. We are actively removed any entries from the dataset that are coded industrial or commercial.

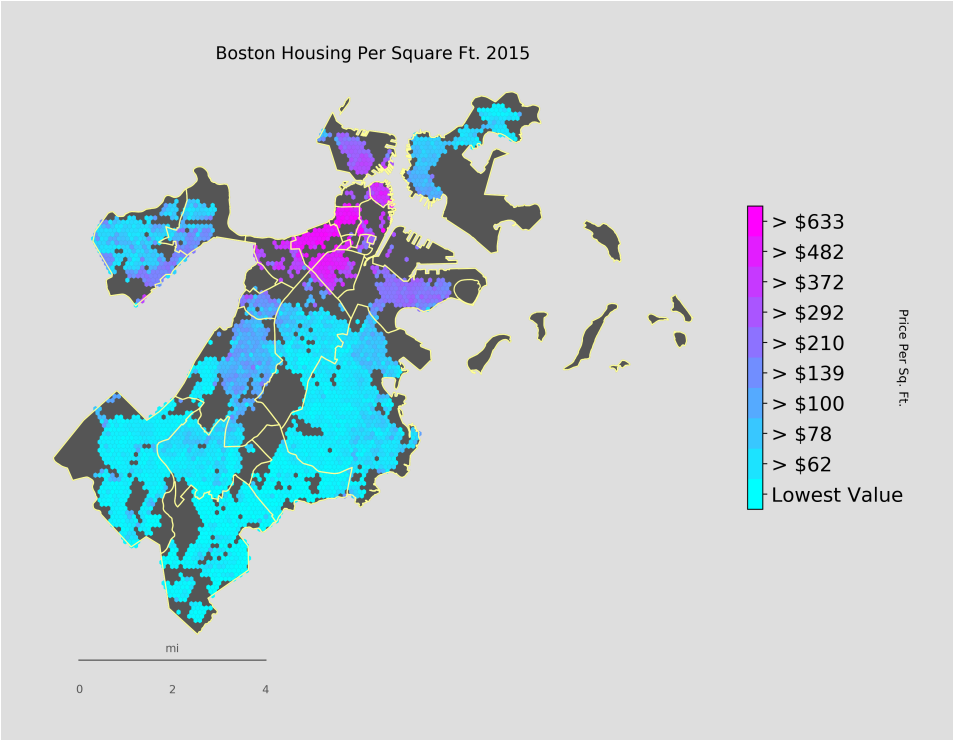

Figure 1: Map of Boston showing the average sq. ft. value of residences with each hexagonal area. Boston proper is a dark gray background while Neighborhood boundaries are shown as yellow lines

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

1 Data and Preprocessing

As mentioned above, the dataset used is provided by Analyze Boston. The data is comprehensive, but incomplete. As such a huge portion of the work put into this project has been in preparing that data for consumption by the neural network. Below is a list of our input vectors and a summary of the work done (if any) during preprocessing.

1.1 Input Vectors for Analysis

- **Latitude and Longitude**:

  Latitude and longitude are ideal for our analysis. Unfortunately, it is only available in a subset of the yearly data sets. To recover the lat/lon data for the remaining years, we applied two different methods:
  - We hooked up to external APIs from Google and the Census bureau and retrieved latitude and longitude using a street address and zip code.
  - Unfortunately, our dataset far exceeded the number of free requests we could make, so for additional records we had to collate data from other years. We took the most complete data set that included lat/lon (2015) and, using the street address as the key, applied those same values to the datasets that did not include them in the first place.

- **Year Built**:

  The year the structure was originally fabricated.

- **Square Feet**:

  A vector easily utilized normalized property values. There is some inconsistency in this area across the dataset, mostly where square footage could either be considered as total land use vs property interior. However this is difficult to determine without going case by case, so rather than trying to alter this data in some way, we're relying on the magic of the network to figure out how to weight it appropriately.

- **Type**:

  This is a string value identifying whether the building is a home or condominium/apartment. This value drives several of our data-cleaning initiatives:
  - While the datasets are broken down at the granularity of 1 row per unit, one of our target vectors, land value, is not. In the cases land value is stored as a single entry for the entire building and each unit in the property is given a land value of 0. While it might make sense to the casual viewer, we've found we submit the data to the neural network in this format, as it creates a very unreal assumption about the actual land value in the given area. As such we've had to fill in the data ourselves, by distributing the total value for the building across all units.
  - In addition to the above, the fact that a unit is a condo or a residence of its own is a value we want to track. Unfortunately this is expressed in a string vector. To feed this into our system, we decided to convert this into a series of boolean vectors that can be assessed separately.

- **Property Value**:

  This is the value given to the housing unit itself, irrespective of the land it sits on. This does not need any preprocessing. This is one of the targeted fields for our predictions.

- **Land Value**:

  This is the value of the land a property sits on. There's some unusual behavior of this field when dealing with multi-unit structures, so the value of a single parcel often has to be distributed across multiple rows. This is one of the targeted fields for our predictions

- **Beds/Baths**:

  Two vectors that did not need any massaging, clear indicators of property value that are easily normalized.

1.2 Other Data Considerations

The data we're working with is far from the only available data set in this field. Over the course of the project we looked at, and to some extent worked on, the following sources:

- **Zillow/Trulia APIs**: These are extensive, but limited to non-paying users. We decided we could not get enough data from this source to properly train our network.
- **US Census Bureau**: This has wide ranging data info, but it can be very inconsistent, the api is not always responsive. Our rough trial run indicated it would take roughly 30 hours per year of data.
- **Analyze Boston Non-Housing Data**: This is something we'd like to leverage for ongoing work – basically plugging in the location of various kinds of services and establishments so that we might gauge their impact on a neighborhood. Unfortunately, time considerations with collating the data prevented us from covering it at this time.

2 Model Considerations

Our predictive engine is a neural network powered by the `sklearn` and `neupy` libraries in Python. A neural network was chosen in particular because it's the strongest predictive model we've worked with up to this point. Our problem is not an issue of classification, so SVM and Graphical Models would not be as effective in projecting future values.

2.1 Network Layers and Structure

When considering the design of our model we considered several methods for structuring the hidden layers of our neural network. Eventually we settled on two hidden layers with a number of nodes equal to the median of a combination of the input (10) and output (2) vectors. After some further testing, we found that we achieved more accurate results without sacrificing too much performance when we expanded the number of nodes up to 50. Past that point we started to lost value compared to run time

Below is a table of test results gathered from training our model against a single year's worth of data:

Table 1: Hidden Layer Performance

Layers	Nodes	Runtime(seconds)	Error
1	10	212	2.477
1	25	367	3.283
1	50	576	0.491
1	100	854	0.587
2	10,10	289	3.05
2	10,25	432	0.479
2	25,25	601	3.75
2	10,50	593	3.24
2	50,50	843	4.242
3	10,10,10	357	0.528
3	10,25,10	512	0.499
3	10,50,10	646	2.27
3	25,25,25	781	2.142

Interestingly, in addition to the lost performance, you can also see that as we expanded the number of nodes past 50, the overall error rose significantly. In fact there seems to be a sweet spot between 30 and 50 nodes overall where the error is roughly the same. At that point we really can boil it down to a performance consideration.

2.2 CNN vs Traditional Neural Network

When considering that one of our main challenges with this project came with the pre-processing of data, it made sense to look at a convolutional neural network as a possible solution to this issue. CNNs have had great success with image preprocessing when "cleaning" low quality images, and could be a potential solution for managing our lower quality data points.

We ultimately did not use a CNN, for the following reasons:

- Adding multiple hidden layers in the form of a perceptron added undue weight to the network itself without a appreciable increase in accuracy.
- Despite the amount of pre-processing needed, the value change function we are modeling is not incredibly complex, and is well represented by a simpler network.
- Most of the preprocessing required, while work intensive, was easy enough to identify. Clear solutions that could help us create accurate input vectors were found in each case, meaning we would not have to rely on a CNN to do that work for us, and in fact could not guarantee the CNN would weight items appropriately.

2.3 Node Pruning and Randomization

Node pruning and randomization are two items we would have liked to have implemented as a part of our model, but are not available at this time in the `sklearn` and `neupy` libraries we've relied on to build our neural network.

In our design scheme, we can easily see how randomization could be very helpful to avoid improper weighting of our internal nodes corresponding to the input vectors. When we have multiple input layers that are related and share a similar relationship to property value (for example, a high value property is likely to have both a large square footage and a greater number of bedrooms and bathrooms), it could be difficult to weight these items correctly.

Similarly, pruning could be a very useful tool, allowing us to remove nodes that apply a marginal value to a network (for example a node that originally weighted bathrooms heavily becomes marginalized as the network learns bedrooms are a greater driver of value.) Re-imagined in a different tech stack, both of these techniques would be incorporated into our model.

2.4 Overfitting

Overfitting is something we have to be careful to avoid with our project, and is also one of the reasons a large dataset spanning multiple years is necessary. Since in many cases property values will not change appreciably from one year to the next and most of the other details of a home will remain the same we'll end up with a number of near-duplicate records in our dataset. This can lead to the network avoiding many value-driving factors such as square footage, condition, etc. and instead applying far too much weight to the location itself (essentially just memorizing location and value combinations.)

It's possible we encountered some of this during our earlier testing, when we were seeing alarmingly low error rates, roughly 10^{-4}. At the time we were working with a much smaller data set (it only spanned a single year), and had done very little preprocessing on the data before training the network. We believe that the lack of year-over-year data, the limited sample, and the "unclean" data vectors led to a case of overfitting. This was ultimately resolved when we expanded our dataset and dedicated resources to preprocessing the data into a form that the network could interpret cleanly.

3 Results

It's not difficult to notice that housing prices are on the rise in Boston, anyone who has lived in the city over the past several years can tell you that. What we've been aiming to do from the start is to identify some of the driving forces behind those changes, and hopefully identify some of the features that add more value to a home or property. Figure 2, below, visualizes the average changes in price/Sq.Ft. in a given hexagonal boundary from 2008 to 2015. This is a rather interesting time period for housing because it begins around the time of the Great Recession and ends in the middle

of a recovery period. As shown, there are both areas where the prices have not only recovered, but outstripped their previous peaks, and those that were still depressed as of 2015. This illustration allowed us to choose representative properties in areas of Boston that have diverse price trends and compare them to the overall price value shown in Figure 1.

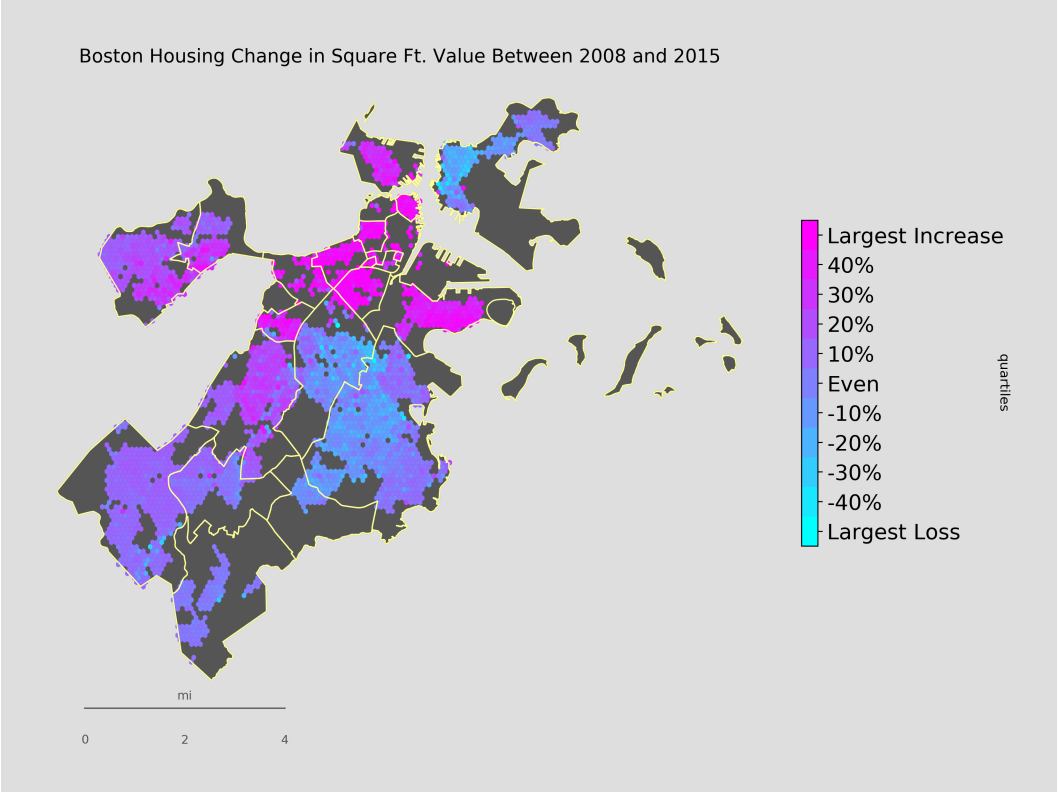

Figure 2: Map of Boston showing the changes in average sq. ft. value of residences with each hexagonal area. Boston proper is a dark gray background while Neighborhood boundaries are shown as yellow lines. The legend shows the quartile breakdown of the property value changes with the light purple color representing no change on average in that area.

3.1 Scope

Once we selected the areas of interest we looked at the availability within our datasets. In order to create a targeted result set that we can discuss in more real terms, we've gathered a subset of properties from around the city of Boston that fit the follow criteria:

- Each property selected appear in the years 2008, 2009, 2010, 2011, 2013, 2015, 2018 (years for which we have the most complete data)
- The properties are distributed evenly over the available zip codes in our dataset
- The properties represent both small and large home (based on square footage and number of rooms)

These criteria allowed us to choose several fictional homes to our model in the years 2019-2021. Once we had extracted this set of properties we were able to not only view the change in value over a set amount of time that has already elapsed, but also project the values out into future years. In order to project we needed to assume that the vectors used for predictions will remain the same (since buildings and properties tend not to change frequently over time) with the exception of year.

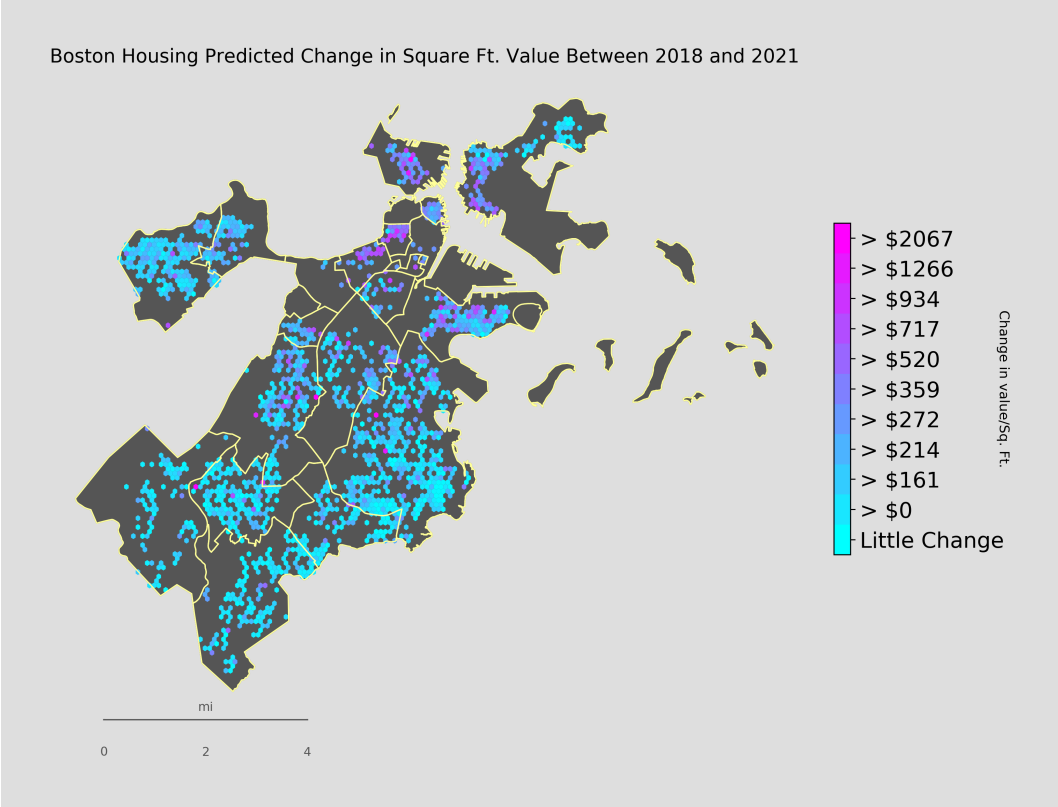

Figure 3: Map of Boston predicting changes in average sq. ft. value of residences with each hexagonal area between the years 2018 and 2021. These properties represent a subset of homes that appear in all years analyzed. The legend shows the quartile breakdown of the dollar property value increase.

3.2 Individual Properties

From the subset of data above in Figure 3, we've taken a deeper dive and extracted a small number of individual properties we can treat as case studies in different zip codes around the city of Boston. Below is a mapping of each address and their change in value of the years. The number in each cell is the total value of a property (Land + Building), and the projected years 2019-2021 are based on a training set from the years 2008-2018.

4 Analysis

4.1 Existing Data

Before getting into the data on a larger scale, it is helpful to review the smaller data set presented in the chart above. If we review the data without the predictions provided by our model, we can see that the change in price over time follows a somewhat consistent curve, declining moderately from the years 2008 into 2013, then rising fairly steeply from 2015 to 2018.

The one notable exception to this is the property located at 28-32 Atlantic Avenue, which experienced an earlier jump in value than the rest of the table, then declined moderately between 2015 and 2018. It is impossible to say for certain the reason for this aberrant behavior, but there is one condition that should be considered. This property is the only condominium in the table, so the property value changes do not include land value, and land value is most likely going to represent the greatest change in value year to year barring active development at a location.

Table 2: Select Property Values

Address	2008	2009	2010	2011	2013	2015	2018	2019	2020	2021
257 255 EVERETT ST 02128	241600	210600	207300	205200	193500	232400	276600	609062	629867	651045
18 R POLK ST 02129	247300	235900	190900	189000	186600	211700	302300	1017543	1049010	1080373
81 83 N MARGIN ST 02113	660200	640500	620600	614400	600700	690200	1051500	851715	881888	912169
536 538 COMMERCIAL ST 02109	759000	759000	717000	701500	798439	1287800	3128345	1284773	1326528	1367674
28 32 ATLANTIC AV 02110	925100	925100	917600	917600	974700	1344304	1262200	731392	756327	781526
234 236 JAMAICAWAY ST 02130	834670	754110	730200	722900	749900	909600	1124700	1195962	1236108	1275906
132 134 HOMESTEAD ST 02121	437570	334970	234100	217700	223200	262500	414100	522132	540259	558813
25 R MAYFIELD ST 02125	415100	395800	335600	332200	306500	343900	440500	433714	449055	464855
325 327 SAVIN HILL AV 02125	490000	442000	408900	404800	429500	468600	616400	539850	558577	577666
86 88 BERNARD ST 02124	436800	349500	300600	282600	277300	317500	364600	453652	469589	486987
47 49 CLARKWOOD ST 02126	444700	343000	286200	283300	299500	346000	463500	396518	410324	424586

4.2 Individual Predictions And Analysis

Before looking at distinct properties, we can quickly identify another consistent trend by looking at the predicted values for the year 2019. The changes from 2018 to 2019 tend to be more dramatic than any other pair of years. This can be attributed to a one time "correction" the network applies, since it is projecting a great deal of value on the neighborhood rather than solely considering the individual property. After this correction, the value changes tend to be a consistent, steady rise.

That said, it's worth looking at the most extreme of these example first, when the property at 18 R Polk Street in the Charlestown neighborhood of Boston apparently triples in value in one year! It seems there are several factors in play. First, in the year 2018 Charlestown is a very desirable place to live and a 4 bedroom 2 bathroom home is a good size for the neighborhood. Zillow estimates its value at 546,000, and its last sale in in 1996 was for 50,000 – what an investment! So why are we past a million dollars in our estimate? The best bet might be to look across the street where a set of luxury condominiums are being installed under the name Lumen Charlestown. A 2 bedroom condominium in this complex is going to run 675,000 dollars, so it's no longer unreasonable to look at the four bedroom house on the same street as a million dollar property.

Next let's look at the property that actually experienced the largest drop in value, 536 Commercial Street in Boston's North End. Zillow currently lists the estimated value at 2.5 million, not exactly the 2018 appraisal, but much closer than our own estimate. So why the huge disparity? Well in addition to a location in a desirable part of the city with lots of amenities, this property stands out as one of the few in the neighborhood with an ocean view, something we have not been able to account for in our model. That said, if you were to consider other similarly sized home in the area, you'd find that a cool million is roughly what you would need to buy a single family home in Boston's North End.

So we've seen how our model is working in a couple of the more well-to-do areas of Boston. But what about a less desirable neighborhood? The first entry in our table covers that exactly, a home in East Boston neighboring the interstate and the airport. This is a small (800 sq ft) home on a small lot in an area that experiences a lot of noise pollution. Nevertheless, our model seems to have taken the long shot and came in with a prediction of over 600,000 dollars, more than twice the value estimate in 2018.

What gives? Well, lets first compare with our friendly third party, Zillow, who gives this home a rough valuation of 432,603, so they seem to be straddling the fence between the historic assessments and our valuations. If we look at the location on the map, we can see that this home is actually closer to parts of downtown Boston (despite being separated by the harbor) than it is to parts of East Boston. Additionally its proximity to the airport means there is a huge black hole of housing data in its immediate vicinity. So what's happening here? Our model has had to look further out to find similar homes with which to make a prediction, and when it did, because we use latitude longitude as our geographic keys, it went looking as the crow flies rather than as the car drives. It started comparing this property to homes in downtown Boston and the Seaport District across the water, both of which claim higher properties values and higher overall growth in value.

Explaining away the various outliers in our data does not mean we should not try to avoid them, but at least helps us understand the challenges we might face when trying to predict values in a complex system that could be represented by any city. It does help warn us that our predictive powers should be looked at on the larger scale when possible, since individual examples can easily fall apart under scrutiny.

5 Conclusion

This project has shown the limitations and potentials for machine learning on publicly available dataset that are maintained by government and nonprofit agencies who's main thrust is not data analysis. The datasets used in this project were purposefully available so that anyone with an Internet connection and the desire to look at the many facets of Boston could do so. But because they were maintained and gifted by many different agencies and companies, there was little consistency across even within the same dataset. Even to the point where a zero value was represented by anything from the number zero as a float, integer, or string, to something as interesting as the letter O. Relative to datasets available on Kaggle that are fully cleaned and meant as benchmarks for developing more efficient machine learning algorithms, the Analyze Boston datasets are meant as a starting point to explore interesting changes throughout the city. They are not meant to be able to be used for a commercial application or product.

But even with all of the aformentioned hurdles, we were able to take the available Boston Housing Value data, distill it down to a fairly consistent subset of properties, and build a predictive neural network to show not only changes in individual houses, but also allowed us to visualize the data in an intuitive way to find areas that are quickly expanding compared to more depressed areas. Finding the intersection of these categories could lead city researchers to areas where neighborhoods are gentrifying and increasing in property value so quickly compared to the income of the residents that vulnerable populations may be pushed out the area.

This could then lead to an increase in certain initiatives in the city has already implemented partially, such as Boston requiring condo developers in the Seaport to earmark a certain percentage of their units to low income residents. This in depth analysis possible with the datasets we used, but only on an overall-trend basis. In order to look at specific areas in depth, it will be necessary to have better starting data that is available from websites such as Zillow, as long as you're willing to pay for it. For now, our model is a great starting point that allows us to find areas for that deserve further exploration.

References, Resources, and Libraries

[1] Analyze Boston-Knight Foundation https://data.boston.gov/

[2] Sensitive Cities, https://sensitivecities.com/so-youd-like-to-make-a-map-using-python-EN.html

[3] Neupy Tutorials, http://neupy.com/docs/tutorials.html

[4] Sci-Kit Learn, http://scikit-learn.org/stable/

[5] Numpy, https://docs.scipy.org/doc/numpy/user/quickstart.html

[6] PyPi Descartes, https://pypi.org/project/descartes/

[7] PyPi Fiona, https://pypi.org/project/Fiona/

[8] PyPi CensusGeocode, https://pypi.org/project/censusgeocode/

[9] GeoPy, https://geopy.readthedocs.io/en/stable/

[10] Pandas, https://pandas.pydata.org/

[11] Zillow Real Estate, https://www.zillow.com/