Boston real estate prediction - motivation

May 1st 2018

Milestone #4 Prepared by Group #8

Paul von Chamier, paul_von_chamier@hks18.harvard.edu
Srikanth Namuduri, srikanth.namuduri@gmail.com
Steven Bowley bowleysj@hotmail.co.uk

Zillow is an online real estate database company. It's "Zestimate" tool uses machine learning and statistical models to analyze property data and estimate home values. It is widely used tool by both buyers and sellers of real estate across the US.

The factors building the "Zestimate" provide a valuable insight into the dynamics of the market and elicit insights about the direction in which various neighborhoods are developing. In 2017 Zillow announced a data science competition aimed at improving the performance of the "Zestimate" based on Zillow data from 3 counties in California (LA, Orange and Ventura). Available information on real estate were used to predict the transaction price. The competition results, published online, provide a robust framework for real estate market trends analysis elsewhere.

The framework and Zillow data can be used to predict real estate market trends in the Boston area. Not only can we build a tool to predict price dynamics on the market and value of a specific piece of real estate but also foresee which neighborhoods are at the greatest risk of gentrification. These insights could inform policy-making on housing by the municipalities. It could also be harnessed by construction companies and investors to guide their real estate projects choice.

Our goal in the project is to use data on real estate characteristics and transaction prices to build a model predicting potential transaction value of real estate in the Boston area. We are applying our model and visualization technique to build a predictive tool for people who are trying to assess a price of a listing now or in the future.

Additionally, we use our quantitative findings to create a heatmap of neighborhoods most likely to experience gentrification in the upcoming years. This tool can allow people to predict future trends and prepare vulnerable communities for the ongoing market processes.

Please explore baseline results of our project using
the interactive tool we created for you:
https://arcg.is/1eDLfX

Introduction and description of data

The data from the website Zillow was imported into the .csv format and processed to produce the initial EDA insights and prepare the foundation for the price prediction model.

Key Variable Name	Description
Address	Address of the property
Age	Age of the property
Agent Name	Name of the real estate agent
Area	Neighborhood in which the property is located
Baths	Number of baths
Beds	Number of beds
City	City limits in which the property exists
Cooling	type of cooling
Garage	Number of garages
Heating	Type of heating
Elementary School	Elementary school for that district
High School	High School for the district
Junior High School	Junior High School for the district
Level	Floor of the apartment
List date	Date when the property was listed
List Price	The listed price
Lot Size	Size of the lot
PhotoURL	URL for seeing the pics of the property
Remarks	Notes from the poster
Sold date	Date of sale
Sold Price	Price of the sale. The predicted variable
SQFT	Square footage of the property
STYLE	style or type of the property
Zip	zip code of the property

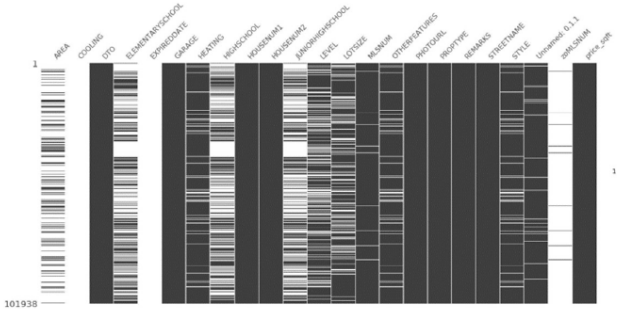
The data at our disposal has a diverse range of types. Let us collapse the long list of available variables showed on the left into types of data points:

Variable Type	Examples
Numerical (Float/Int)	Age, Area, Beds, Baths
Strings	Address,
Date-time	Sold date
lat-long	Latitude and Longitude - geographic coordinates
Image	Google Street View Images of the property

This diverse range gives a strong base for a model. It allows us to look not only into trends across properties but also across time. We used the following methods to explore data:

1. Visualizations (presented on the next 2 pages)
   - This includes histograms, scatterplots, geocoded maps and price heatmaps.
2. Regressions
   - We have started by having the PCA model clustering analysis and simple correlation regressions to finds the most obvious patterns.
3. Data wrangling and cleaning
   - We worked through the data to identify the missingness and averages across categories.

Identified variables with missing values. White color denotes the missing proportion of data.



The data comes from the internet real estate overview platform Zillow. It covers 500 000 positions which is enough to identify big data patterns in the area. 160 000 include the transaction data (They were sold). The data was collected using scarping of publicly available listings.

Upon assembling the data, we transformed the addresses of the Massachusetts listings into geocoded coordinates and created a map:

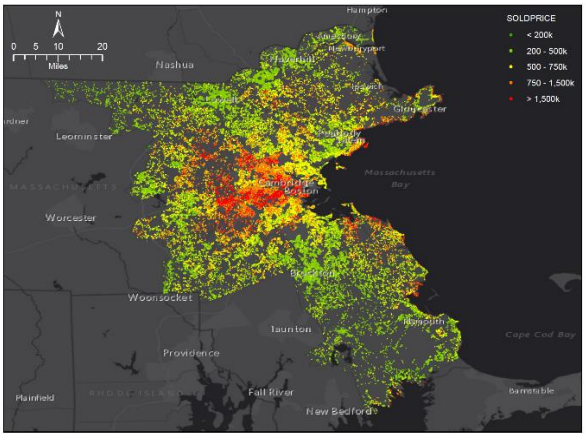
Figure 1. Geographic distribution of available listings

Source: own elaboration

We can clearly tell that the data is covering not only the Greater Boston Area but actually the entirety of Massachusetts. This initial insight suggests we trim down our dataset to cover only Greater Boston Area.

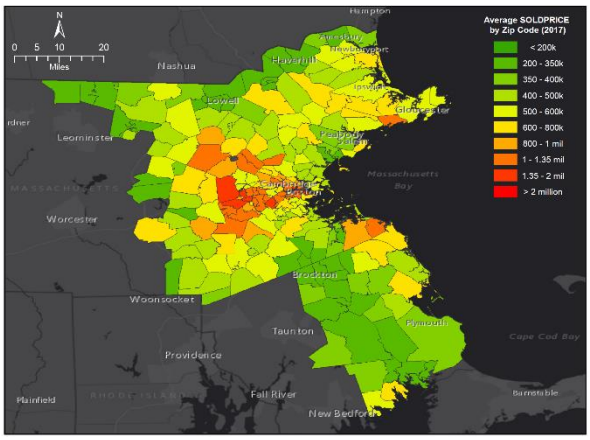
Using official census data as well as the external database of City-data[1], we managed to identify 231 ZIP codes, which encompass the Greater Boston Area. Below is the zoom in on the sales prices in Greater Boston Area – individual ones, and the average per ZIP address:

Figure 2. Sell prices in Greater Boston



Source: own elaboration

Figure 3. Average sell prices in Greater Boston per ZIP



Source: own elaboration

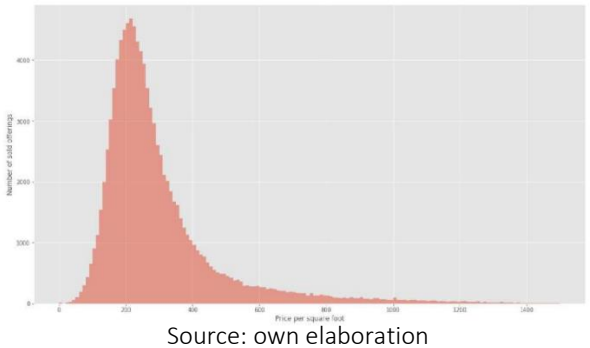
The Greater Boston market is very dynamic and prediction of real estate prices is a key competence for both sellers and buyers alike. Our analysis provides tools to predict the market value of a listing a inform investment choices.

[1] http://www.city-data.com/zipmaps/Boston-Massachusetts.html

The fancy residential areas on the downtown peninsula follow. Then there is a layer of neighborhoods around that area with somewhat high prices. However, it is no comparison to the Peninsula. The rest of the Greater Boston is a different league – property is way cheaper there.

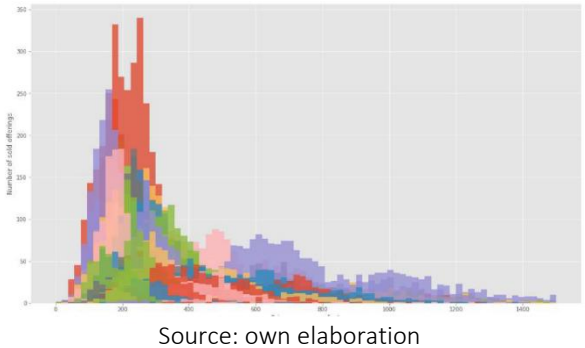
The overall average price of a square foot in the Greater Boston Area is 303.60$. The distribution reveals an asymmetrical distribution with a long "tail" towards expensive listings. This represents the downtown peninsula. We will now break down the distribution per zip address. Below on the right side we see that the distribution is similar for various neighborhoods but for some of them the tail is thicker and tilted to the right. Those are expensive neighborhoods.

Figure 4. Histogram of square foot prices



Source: own elaboration

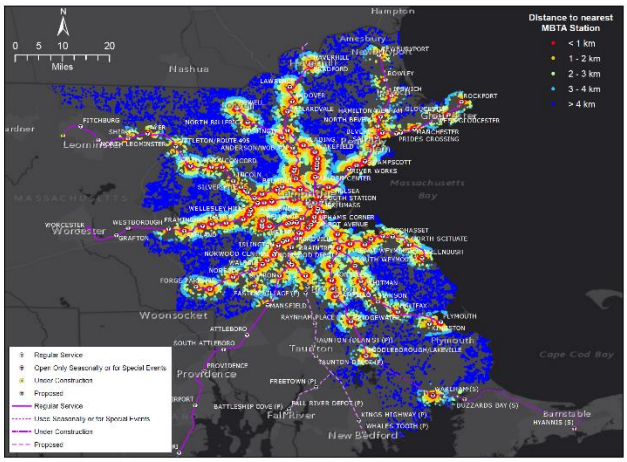
Figure 5. Histogram of square foot prices per ZIP



Source: own elaboration

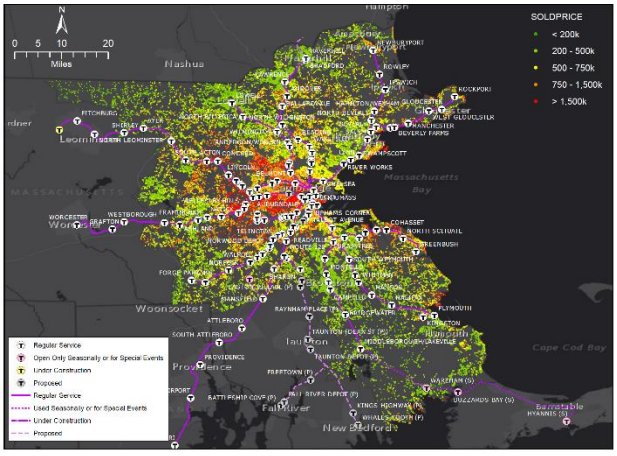
To further enhance the quality of our data, we added an external data type to our dataset – distance from the closest MBTA station. Access to public transportation might be a relevant factor in buying a piece of real estate. Below you can see our catchment area assessment and the superimposition of the network against the average prices on the map.

Figure 6. MBTA Catchment area assessment



Source: own elaboration

Figure 7. MBTA network and real estate prices



Source: own elaboration

Literature review

Below are the positions which influenced our approach to the project:

1. Zillow Kaggle Competition: https://www.kaggle.com/c/zillow-prize-1

2. Using Python to scrape Google Street images: https://andrewpwheeler.wordpress.com/2015/12/28/usingpython-to-grab-google-street-view-imagery/

3. A. V. Dorogush, A. Gulin, G. Gusev, N. Kazeev, L. Ostroumova Prokhorenkova, A. Vorobev, Fighting biases with dynamic boosting, CoRR, 2017.

4. T. Chen and C. Guestrin, XGBoost: A Scalable Tree Boosting System, In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). ACM, New York, NY, USA, 785-794, 2016.

5. Q. You, R. Pang, L. Cao and J. Luo, Image-Based Appraisal of Real Estate Properties, in IEEE Transactions on Multimedia, vol. 19, no. 12, pp. 2751-2759, 2017.

Reference 1: Zillow Kaggle Competition

https://www.kaggle.com/c/zillow-prize-1

Zillow is an online real estate database company. It's "Zestimate" tool uses machine learning and statistical models to analyze property data and estimate home values. The "Zillow Prize" is a $1 million competition hosted on Kaggle to improve the accuracy of the Zestimates. Participants are developing algorithms to predict the future sale price of homes. The first round goal was to build a model that reduces the Zestimate residual error (submission 10/16/2017). The second round is to build a complete home valuation algorithm. Results are evaluated against three month sales after the deadlines. Submissions should predict the logerror = log(Zestimate) – log(SalePrice) for various time points. The data provided includes all information on properties in 3 counties in California (LA, Orange and Ventura) and their home features for 2016/2017. Training data includes the actual transaction prices, with test data hosted on the Kaggle for calculating the leaderboard.

Reference 2: Using Python to scrape Google Street images

https://andrewpwheeler.wordpress.com/2015/12/28/using-python-to-grab-google-street-view-imagery/

The Google Street View Image API allows you to scrape the latest street view image by entering a set of coordinates or an address. Andrew Wheeler provides an example of a function in python, which takes an address and location for the image download. Google includes date/month of the image, however only the latest images are currently accessible unless dates of earlier images are known.

Reference 3: Fighting biases with dynamic boosting

Gradient boosting is a powerful machine learning technique, useful for analyzing noisy, complex data. This paper identifies and solves the problem of bias (towards the training sample) in pointwise gradient estimates, which can lead to over fitted models. It proposes a dynamic boosting approach to ensure unbiased residuals for regression, at a minimal cost to the variance of the gradient estimation. The proposed algorithm is computationally inefficient, however optimization tricks are suggested. The paper states that the dynamic boosting implementation outperforms XGBoost and other popular gradient boosted decision trees. The article presents gradient boosting method, estimation of pointwise regression gradients, and the impact of training bias on gradient estimation. It then presents the dynamic boosting method to combat gradient bias. This computationally inefficient method iteratively constructs a model by training multiple iterations of models to make the residual on the samples unbiased, iteratively creating unbiased residuals at each iteration. Comparisons with baseline algorithms (XGBoost, LightGBM, Soft-Plain) show that empirical results of the dynamic boosting method outperform the baseline packages.

Reference 4: XGBoost: A Scalable Tree Boosting System

XGBoost is a highly effective and widely used open source package that provides a scalable end-to-end tree boosting system. It enhances gradient tree boosting methods or Gradient Boosting Machine (GBM) / Gradient Boosted Regression Trees (GBRT). The article explains how the algorithm is built, and proposes methods for weighting quantiles, data sparsity awareness and cache aware block structures for out of core learning. Characteristics of other major tree boosting systems are compared to XGBoost, including scikit-learn, R GBM, Spark MLLib. XGBoost is presented consensus choice learning method, and as the most state-of-the-art system, given its incorporation of a range of techniques, including out-of-core processing, sparsity awareness, and exact greedy algorithm for data splitting. (17/29 Kaggle Challenge solutions in 2015 used XGBoost). XGBoost is available as a single machine or distributed version which runs on Hadoop. Lessons learned from this paper are that cache access patterns, data compression and sharding are essential elements for building a tree boosting system.

Reference 5: Image-Based Appraisal of Real Estate Properties

A picture is worth a thousand words. Current research indicates real estate value is closely related to property infrastructure, traffic, online reviews, etc. Real estate price indexes and estimators have largely ignored images as a factor, due to difficulty of interpretation, or quantification. Current computational infrastructure is making the analysis of visual content faster, cheaper and more feasible. Deep learning techniques, in particular deep learned visual features, such as those used in Convolutional Neural Networks (CNN) take into consideration the location and values of neighboring image pixels to solve computer vision tasks. This paper presents an approach to using photographs to predict real estate prices. The method uses the random walk technique to generate house sequences according to physical location of each house, and Recurrent Neural Network (RNN) for prediction. The model is a Multi-layer bidirectional Recurrent Neural Network (BRNN), trained using bidirectional Long Short Term Memory (B-LSTM). Data used in this study included pictures, and geo-location was obtained from the Microsoft-Bing Map API to provide address geocoding (latitude and longitude). Vicinity distances between properties were calculated based on coordinates. The implementation used the GoogleNet deep neural architecture, and compared it with other algorithms (LASSO Regression, DeepWalk and RNN. The article concludes that location and visual attributes from pictures can be used to effectively predict house prices, using the proposed deep learning approach.

Modeling approach

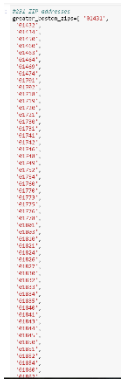
Our baseline mode is the Neural network-based model. Beyond that, we also used the XGBOOST Tree Boosting system as an additional robustness check for the results of the baseline model. Below you can analyze the steps we have taken to come up with the predictive model

1. Destringing the data
   - Many of the columns included multiple data points separated by a comma or a space, in the first step we split them and organized into separate valuable features.
   - At this initial step random values with only single observations and erroneous values (e.g. 1000 beds in a listing) were verified.

Basement	Fireplaces	Roof	Floor	Appliances
No	2	Asphalt/Fiberglass Shingles	Tile, Wall to Wall Carpet, Concrete	Dishwasher, Disposal, Microwave, Refrigerator,...
No	0	Asphalt/Fiberglass Shingles, Rubber	Wall to Wall Carpet, Concrete, Hardwood	Range, Dishwasher, Disposal, Microwave, Refrig...
Yes	1	Asphalt/Fiberglass Shingles	Tile, Wall to Wall Carpet, Concrete, Hardwood	Range, Dishwasher, Disposal, Microwave, Refrig...

bamboo	carpet	carpets	concrete	engineered	flooring	green	hardwood	labeled	laminate	...
0	1	0	1	0	0	0	0	0	0	...
0	1	0	1	0	0	0	1	0	0	...
0	1	0	1	0	0	0	1	0	0	...
0	0	0	0	0	0	0	0	0	0	...
1	0	0	0	0	0	0	0	0	0	...

2. Transforming categorical variables with thousands of unique values into binary ranges
   - Variables with hundreds or thousands of unique observations (e.g. elementary school districts) were categorized into 25 clusters according to their average sell price. As an effect we retain control over sparseness of the dataset we are crafting.

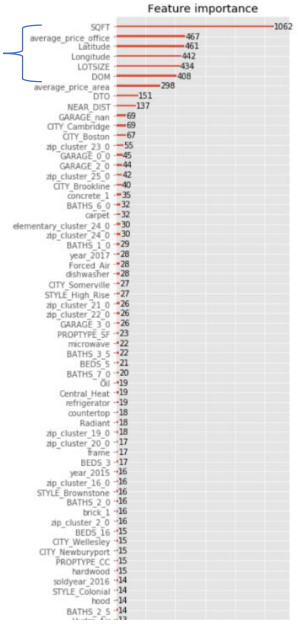


```
Out[510]: ZIP                225
          average_price_zip  225
          zip_cluster         25
          dtype: int64
```

3. Removing missingness using regression with random noise
   - We used complete observations to predict the missing values. Before imputing them we add random noise value to each one of them not to bias our dataset in any direction.
4. Normalization of the dataset
   - Our dataset has columns measuring values in hundreds of thousands next to binary values and columns measuring time in months and years. This might harm the accuracy of the model. For this reason, we normalized each column between 0 and 1 not to bias the model towards any of the columns we are using.

5. Random sampling of the data into train and validation datasets
   - We randomly split the data into the train and test datasets. The models are fitted using the train dataset while the final accuracy will be reported using the validation and test dataset.
6. Neural Network Model
   - We ran the Neural Network Model with the following specification
   - The achieved accuracy is 100% accuracy if a listing has a listing price and 81% prediction accuracy just by looking into qualitative features of a house (ignoring the listing price)
7. XGBOOST Model
   - We ran the XGBOOST Model as a robustness check. The result was similar to the one achieved by the neural network. The accuracy was 79.5%.
   - We used Grid Search Cross Validation to enhance the accuracy.
   - The XGBOOST analysis allowed us to gain an insight into what features are mostly driving the prediction. Here is the break-down:

Top 5



8. Neural Networks:
   - We first fit a very simple model with 20 nodes in the first layer and only 1 hidden layer. This approach gave us a test R square of 77%. Then we experimented with various configurations of neural networks with nodes in the first layer between 20 and 250. The number of hidden layers was varied between 1 and 8. We varied the regularization parameter, batch size and the number of epochs to further optimize the results. Even with the biggest and deepest network we tried, we got a test R Square of 80%. The best result we got from XGboost was 78%. So the best result we have is with the neural networks - even though the margin is not very big.
9. Final Verification on the test dataset
   - We used the January 2018 data provided for the test verification. We received similar results for Sell price prediction – 100% with the list price and 81% without it. Regarding the DOM prediction, we received 71% prediction accuracy.
10. Gentrification analysis
- As an additional contribution, we used the price dynamics and our predictive model to analyze the gentrification patterns in the Greater Boston Area.

Project trajectory, results, interpretation

1. Drawing plans

   Our initial ambition was to enhance the model with three things:
   - Transform addresses into geocoded coordinates
   - Use photo URLs to run neural network on picture features
   - Add depth to the dataset by adding a column on MBTA catchment area distance

   Soon enough we realized that running geocoding queries with Google and other providers is creating a problem of a daily limit of requests. We had around 100,000 observations to go through but Google would refuse to share coordinates after a few hundred a day, and that with many timeout errors. We needed to change the approach. We downloaded the database of Massachusetts coordinates and run a query on a local machine. This worked. We then run our regressions using individual listings as well as by grouping them by ZIP addresses or by US census areas, which are smaller than ZIP addresses.

   In terms of photo URLs, the trouble was to download pictures for 100,000 listings. Web connection would crash and the photos would be of varying quality. We managed to download a few thousand photos though. Some of them were showing interior while the others were showing side of the house or the whole street. We would have to create a few neural networks models before being able to translate that into an actual insight for price prediction. The GPU available on Jupyter Hub was too limited. We took a strategic decision to concentrate on the next goal.

   We scraped data from MBTA on the coordinates of transit stations. We then run a two-stage model. In the first stage each listing was allocated to the closest station. In the second stage the distance to that station was assessed. The result was an additional variable we used for prediction. We also created catchment area visualization thanks to that.

2. Reducing dimensionality

   Many categorical variables had thousands of unique values. We solved it by grouping them into 25 clusters defined by sell price range (as described in the modeling approach).

3. Modeling

   The 100% accuracy if we include the listprice and 81% if we don't use it is a result of a long process of adjusting neural network parameters to accommodate a trade-off. The trade-off was that more intricate neural networks would not converge with accuracy within the given number of epochs which in turn was limited by the GPU capacity. Eventually we stroke a balance between the learning pace of the model and the GPU capacity.

   The 81% accuracy reflects two trends according to our interpretation:

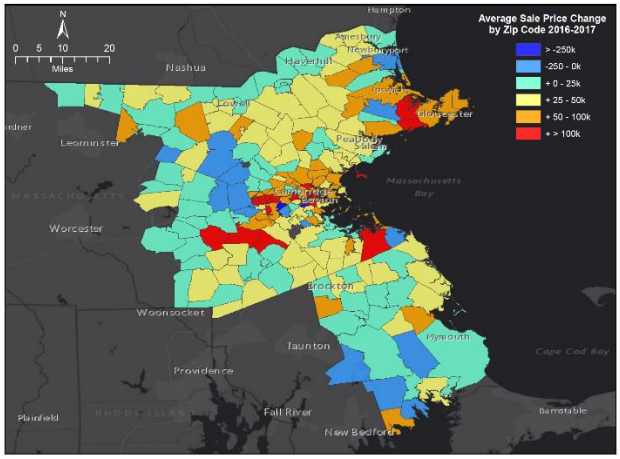
   1. There are more features which are not present like ethnic minorities clustering together, buyers/sellers anticipating future neighborhood trends, quality of the interior materials, the view or the above sea level (Climate change risk factor).
   2. Time constraint – Buyers are constrained by the time window they choose or get from their bank to make a purchase. They don't have a whole palette of listings from 2-3 years like we had for the analysis. They only have currently available listing and only in their areas of interest. This might create an element of randomness in the sell prices.

3. The final modeling insight is that Square feet are by far the most important price factor. They are followed by the average price per realtor office (They seem to specialize in certain price ranges), Latitude and longitude (location), The lot size and the DOM (Days on Market) – pricier offers take longer to sell.

4. Gentrification

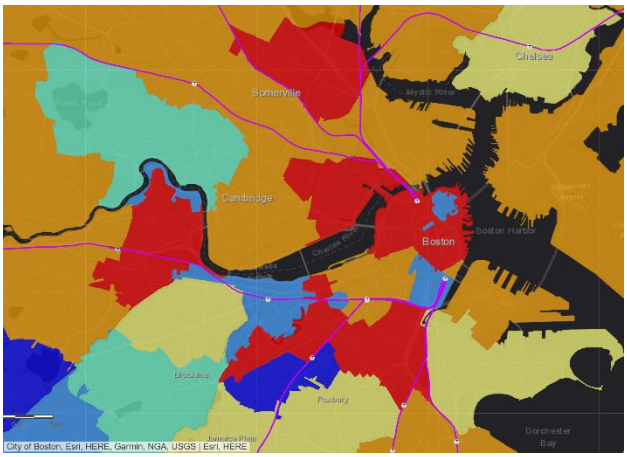
   We used the predictive model and the geocoding to come up with the most gentrifying neighborhoods. Here are the heatmaps:

Figure 8. Gentrification in Greater Boston



Source: own elaboration

Figure 9. Zoom in on Boston proper



Source: own elaboration

Somerville and Allston are gentrifying aggressively. The same happens to East Cambridge (Kendall/MIT T stop). The same happens to parts of downtown Boston. Interestingly we also see a reverse process unfolding in some areas. Blue areas denote neighbors where average price predictions have been actually falling over the last 2 years despite the city-wide trend.

Conclusion and future ideas

Publicly available data, present at various sources simultaneously (Zillow, Census data, MBTA,..) can be combined to devise a powerful predictive mechanism for price dynamics. As we have shown, the sell price predictive power is very close to 100% for listings with a price and more than 81% just by looking at qualitative features of the listing. For the Days on Market (DOM), we can predict them with accuracy of more than 73% (or, again, close to 100% if the list price is present).

The model can be used by both sellers and buyers to inform their investment choices. The communities might use it to anticipate gentrification trends and address them ahead of time.

There is space for further growth. In the future, with more computational power and a more robust internet connection it will be possible to scrape photos for all the hundreds of thousands of listings and run the CNN – neural networks aimed at identifying visual patterns in the pictures. This would allow us to identify visual features of listings at various price levels and further improve the prediction. A second idea is to use data on ethno-cultural composition of a neighborhood to address predict price preferences of specific buyers.