Apartment-Reduce: An Exploration of Fraud and Value in New York City Apartment Listings

Yijie Feng
New York University
New York, NY
yijiefeng@nyu.edu

Vignesh Kamath
New York University
New York, NY
vk970@nyu.edu

Omer Solmazer
New York University
New York, NY
os788@nyu.edu

Abstract

Our project analyzes a dataset of New York City apartment listings pulled from the popular real estate listings sites Craigslist, StreetEasy, and RentHop to distinguish between spam/fraudulent listings and legitimately good values. The goal of this project is to make the decision making process easier for prospective renters in New York City by extracting only meaningful and legitimate information from a large collection of listings and classifying neighborhoods by price and value.

In our analysis, we use a quality measure for apartment listings, defined as a function of a listing's price, space, neighborhood, and features. From these scores, we can classify listings in our dataset and group them based on what range of scores they fall under (spam, high value, and low value categories will have associated ranges).

I. Introduction

For prospective renters in New York City, the easiest way to research apartments is by browsing real estate listings websites. Unfortunately, in a city as large as New York, this process can be long and cumbersome, with fake listings, scams, and irrelevant results occupying a large amount of space on user-submitted listings websites. Our project Apartment-Reduce takes a look at data from these listings websites to profile the various neighborhoods in New York City and to distinguish between spam/fraudulent listings and legitimately good values. From this project, we hope to develop a system and scoring metric for automatically extracting relevant information from real-estate listings websites.

There are several applications for such a tool, including but not limited to:

- Newcomers to the city can use this tool as an accurate reference point for apartment prices and avoid scams

- Apartment listings sites like Craigslist and Streeteasy can use such a tool to improve the quality of their service
- Law enforcement will be able to use such a tool to track down scammers
- Real estate speculators can use this tool to identify prospective neighborhoods to purchase properties in

Real Estate listings are a great source of data for looking at real estate trends in an area since they contain so many quantifiable features such as price, space, neighborhood, number of bedrooms, as well as characteristics of the listing itself (has images, has description, etc.). We can use these characteristics to score all of the listings in our collection and classify them based on what range of scores they fall under (spam, high value, and low value categories will have associated ranges). By grouping listings by neighborhood, we can get market-value approximations and trends for different neighborhoods.

Using Real Estate listings also makes the evaluation process straightforward. Once we have results, we can evaluate our analytic by categorizing a subset of our results as being true positives, false positives, false negatives, and true negatives. We judge it on qualitative factors such as quality of images, background of poster, etc. to evaluate the correctness of the analytic. While there is no way to prove whether a listing is definitely spam or not, we can use human judgment to support a small subset of the results of our program.

Data Sources:

Data Source	Data Source Description	Data Collection	Data Size	Data Frequency
Data Source 1: Craigslist	Craigslist is one of the primary sources for selling/renting apartments all over New York City. It contains user generated listings, thus the data is scattered and unstructured. Craigslist requires the following fields: postal code, posting title, number of rooms, number of bathrooms. Other features such as laundry, parking, rent price, and square footage are optional for the poster to include. We will also factor in the number of photos and length of descriptions when assigning “quality” scores to the listings	batch collection	(10000 listings) ~ 1MB	We estimate that ~1000 new Craiglist listings are generated per day in the most active NYC neighborhoods. Data collection will occur daily as batch jobs and the crawler will parse the unstructured data to extract price, neighborhood, and apartment characteristics.

Data Source 2: Streeteasy	Streeteasy is an apartment listings site where most of the listings are provided by brokers and apartment owners directly. The data pulled from Streeteasy is well-structured: all Streeteasy listings include the # of rooms, neighborhood, address, and descriptions as well as price histories. Streeteasy is unlikely to include spam since individuals need to pay a fee in order to post listings, but we can use the price history data for listings to uncover good values.	batch collection	(11000 listings) ~ 100KB	We will download new listings data from Streeteasy as several batch (~once per week) jobs for date ranges that don't overlap.
Data Source 3: Renthop	Renthop is an apartment listings site where most of the listings are provided by brokers and apartment owners directly. Thus, the data is well-structured. Renthop allows for user-submitted posts so there might be spam listings, but it most likely not as many as in Craigslist As in Craigslist, we will score listings based on the number of features, number and quality of images, and filter listings for spam before scoring.	batch collection	(100000 listings) ~ 2.5MB	We will download new listings data from Renthop as several batch (~once per week) jobs for date ranges that don't overlap.

II. MOTIVATION

New York City has the highest population density of any county in the United States at 27,812/km². This high population density combined with the abundance of con-artists and predatory brokers in the city makes the apartment hunting process especially painful for newcomers.

Prospective renters often begin their search by using public apartment listings sites or real-estate brokers. Given that demand for housing is at an all-time high, renters in this market are able to take advantage of uninformed buyers to rent rooms at prices above market-value. Our project attempts to address this problem of information asymmetry in the apartment-hunting process by computing up-to-date measures of relative value across the city, factoring in characteristics like neighborhood, bedrooms, and listing quality.

The size and high turnover rate of the New York City real-estate market make it an interesting subject for analysis. The high rate of turnover for apartments in the city allow us to collect a large data set representing properties from all neighborhoods in Manhattan (and popular neighborhoods of Brooklyn) for a very narrow time window. From these listings, we can compute snapshots of market value monthly-rent prices for apartments with particular characteristics. Furthermore, the New York City boroughs of Manhattan and Brooklyn tend to have well-defined neighborhood boundaries with distinct price distributions. In our analysis, we use these figures as starting points when determining price/value mismatch and combine it with a spam filter to filter out unusable data.

III. Related Work

[5] Extracting Structured Data from Web Pages

This paper discusses strategies for automatically extracting structured data from a given set of pages using a generated "template." An important characteristic of pages belonging to the same site is that the data encoding is done in a consistent manner across all the pages, so structured data can be found by using a template T for a schema S. T can be defined as a function that maps each type constructor r τ of S into an ordered set of strings T(τ), such that,

1. If τ is a tuple constructor of order n, T(τ) is an ordered set of n + 1 strings Cτ1,...,Cτ(n+1).
2. If τ is a set constructor, T(τ) is a string Sτ (trivially an ordered set of unit size).

This algorithm (titled ExAlg) works in 2 stages. In the first stage (ECGM), it discovers sets of tokens associated with the same type constructor in the (unknown) template used to create the input pages. In the second stage (Analysis), it uses the above sets to deduce a template. This template is then used to extract values encoded in the pages. The first stage of EXALG (within Sub-module FINDEQ) computes "equivalence classes" — sets of tokens having the same frequency of occurrence in every page in Pe (LFEQs - Large and Frequently occurring EQuivalence classes). In the second stage, it builds an output template TS using the LFEQs constructed in the previous stage by generating a mapping from each type constructor in to ordered set of strings.

In the algorithm evaluation, for 18 or 40% of the input collections the system correctly extracted all the attributes. For other collections there were a small number of attributes that were extracted only partially correctly. Specifically, on an average, around 80% of the attributes were extracted correctly.

[6] Detecting Anomalies using Benford's law

The paper operates on the principle that, within a large enough universe of numbers that were naturally compiled, the first digits of the numbers would occur in a logarithmic pattern. The first digits of numbers are the non-zero, absolute value integers. This statistical oddity provides an opportunity to analyze vast amounts of data for anomalies. If the first digits within the numbers of the data being analyzed does not follow Benford's pattern, then something unnatural has happened with the data.

The basic formula is:

x = log10 of 1+(1/n)

Where n is the digit or combination of digits being tested and x is the percentage of their occurrence. This works under the assumption that the universe of numbers must be large enough for the distribution to take shape. Some have found that a universe smaller than 100 items will not exhibit the pattern. Second, the numbers must be free of artificial limits or origins. The author then shows data of his experiments where he successfully finds anomalies in large sets of data by applying Benford's law using a SAS program. Once the anomalies have been found, they can be looked at together to find the cause.

[7]Big Data: Issues, Challenges, Tools and Good Practices

As the volume of data collected increases, the challenges of storing and processing data are increasing rapidly. Not all this data is structured as was before, with device logs, sensor data, social media data etc., the transforming of the data itself into meaningful subsets is also getting problematic. Since the data flow is very rapid, the processing time should be as short as possible in order to catch up. In order to match the data, common columns should be found or added as meaningful values in order to make the whole data work as unibody.

In our project, we collect data from three different sources and parse through unstructured data on these pages to get the relevant fields. From here we unify the data using neighborhood and bedroom characteristics.

[8]A Survey of Clustering Algorithms for Big Data: Taxonomy and Empirical Analysis

There are numerous different algorithms to cluster data according to similar features that individual elements' have. When processing large datasets, selecting an efficient clustering algorithm is important in order to minimize the time it takes to group and detect outliers. Clustering algorithms are divided into subcategories: Partitioning-based, Hierarchical-based, Density-based, Grid-based and Model-based. Among these different methods, K-means and Birch algorithms stand out from the rest. They both can handle large datasets and are efficient to implement. Later, authors introduce FCM (Fuzzy C(K) Means), an algorithm based on K means but which assigns elements to clusters with a probability. Since they all have their differences, it might be best to implement more than one algorithm and compare the results to each one. This way, the common outliers can be marked with higher confidence of being spam listings and the rest can be marked as a candidate for spam, which yield the most reliable output.

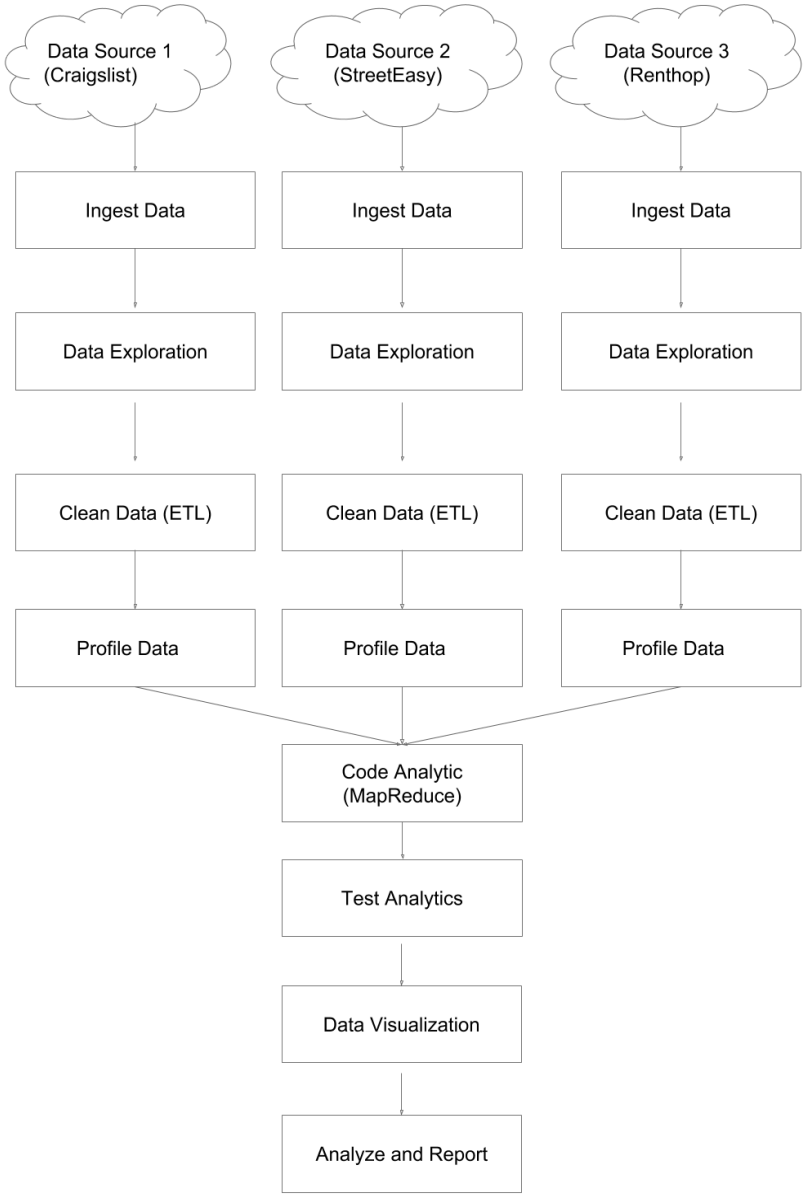
[9] Spam Detection Using Text Clustering

This paper describes a clustering method used to detect spam in a collection of emails -- specifically it talks about a method of identifying spam using text clustering instead of the traditional machine-learning approach. In their method, they compute clusters using the k-means algorithm; each cluster is labeled as either "spam" or "non-spam" based on the proportion of emails in that cluster that is spam. This proportion is the "threshold value" and is a measure of precision for the spam filter. Incoming emails are then assigned to a cluster and labeled as "spam" or "non-spam" based on which cluster they get assigned to. The paper shows that with the correct spam threshold selection (minimum % of spam in a cluster for it to be considered a spam cluster), the text-classification method can outperform a Bayesian spam filter and match the performance of SVM, while using less resources.

[10] A web crawler design for data mining

This paper describes a web crawler program that can effectively download web pages and extract information from those pages. Web crawlers are typically used by search engines to crawl through the web and index pages for search engines to use, but also have applications in data mining. The paper describes a distributed method of web crawling that makes use of a group of machines to increase efficiency and response-time when crawling a large collection of documents. The architecture described consists of a group of machines where processing is assigned to idle machines. There is a single control unit that acts as the driver program and many crawler units that carry out the task of downloading files to permanent storage. While crawlers act independent of each other, they must communicate to reduce redundancy; this communication occurs through the control unit. This distributed architecture allows mining of large datasets while maximizing the available resources in the system (CPU cycles and disk space).

IV. DESIGN



Ingest/Explore Data:

Our dataset is compiled from a set of apartment listings pulled from three apartment listings sites: Craigslist, StreetEasy, and Renthop. For all data sources, data ingestion is done via a web-crawler program that parses useful fields from a collection of pages and outputs these fields as a text-file that conforms to our standard format.

In all of our data sources, the crawler looks the for number of images, description, price, and neighborhood fields. The crawler waits for a specified amount of time between downloads, during which it processes the previously downloaded page.

ETL:

Our ETL process occurs as we crawl our data sources for listings. Since the bottleneck in this system is the download rate, the we are able to more efficiently process listings by directly integrating this step with our crawler, which will produce a structured text file for the profile and analytic phases.

Some of the things we consider are eliminating non-alphanumeric characters, removing newline characters within descriptions, and standardizing neighborhood names and formatting. For our final analytic computation, we refer to this list of known neighborhoods to filter out irrelevant results and compute price statistics per neighborhood.For information that is present in Craigslist listings but not in others, we append the additional information at the end of each line.

Profile:

We run a Hive script to process the dataset of apartment listings in the profile/analyze phase. Here, we group listings by neighborhood and get a basic summary of the distribution of values within the dataset by computing min, max, and average price and apartment characteristics. From our output, we obtained ranges for price, bedrooms, and text length of the descriptions. These ranges are used in our spam scoring formula to normalize values.

Analytic:

A MapReduce job takes the structured text file from the web crawler as input and creates keys consisting of (neighborhood, bedroom) pairs. The reducer filters out spammy listings before proceeding with the calculations for the reduced set of listings. All listings are passed through the following pipeline:

1. The filter first removes all listings that don't contain a valid neighborhood
2. The filter removes all listings with abnormally low prices. We selected $200 as an arbitrary minimum monthly rent for a listing to be considered legitimate.
3. For all listings that pass the first two filters, we use the ranges obtained from the profile phase to assign a score to each listing that designates the likelihood that it is a legit listing (rather than spam):

$$P_{legit}(l) = max(\ 4.5(log(zscore' + 6)) \cdot 2.5((log_{25}(num\_images' + 1)) \cdot ((d\_length')/150) \cdot 2(has\_map'),\ 0\ )$$

This function approximates the probability that a listing is legitimate based on its z-score, the number of images in the listing, the description length, and whether or not it has a map. We assign weights to each listing field based on its importance relative to other fields when detecting spam and normalize its value by dividing by its the max value in the range for that field.

When evaluating a listing, the benefit of having additional images, description length, and high prices tapers off after a certain number. To account for this behavior, we take the log() of each of these fields.

Test:

We evaluate each run of our analytic to fine-tune the weights assigned to each field in P() and the optimal threshold value of P(l) to use when filtering. After each run, we scan the list of identified spam listings to look for false positives in our results. To identify false negative results, we look at a sample of listings taken from the end of our output file.

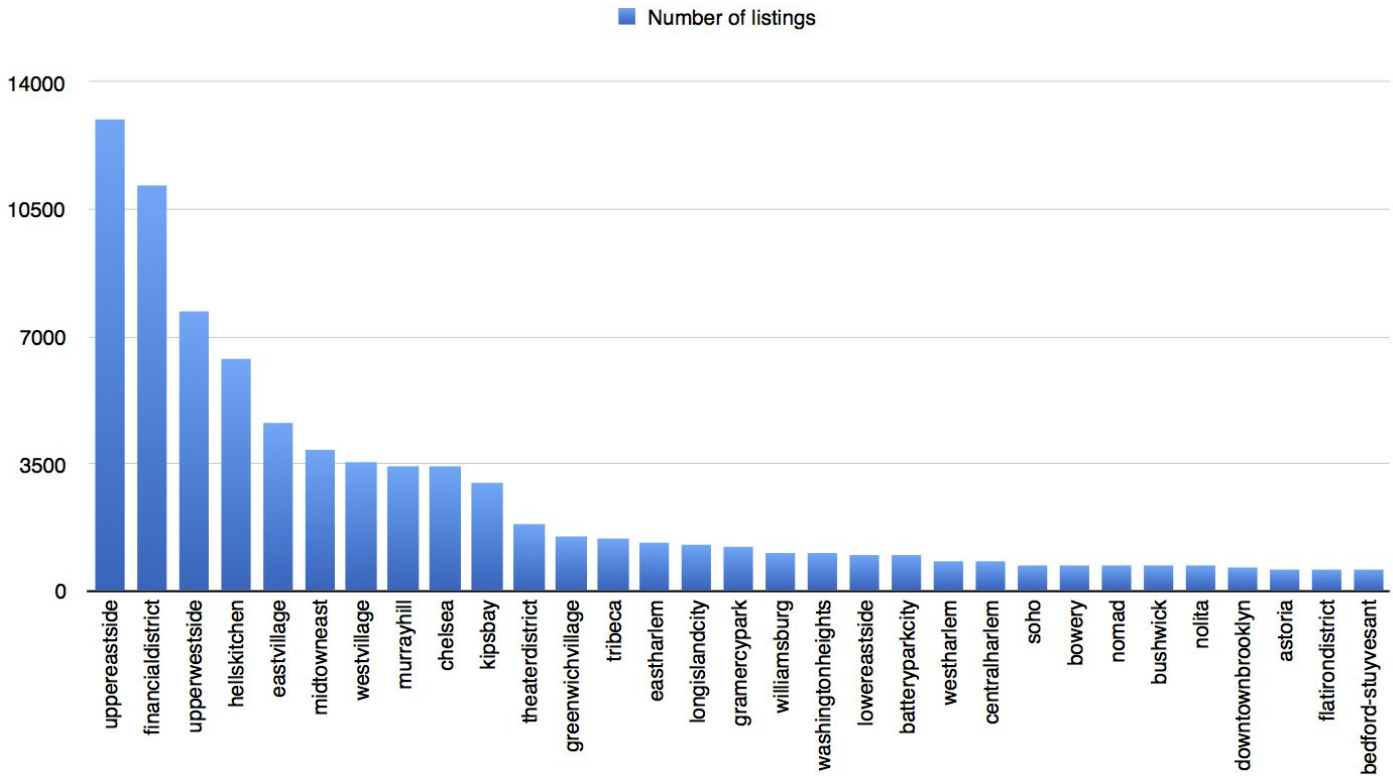
Through trial and error, we approximate the optimal field weights and threshold by adjusting values until we produce the lowest number of (false positives + false negatives).

V. Results

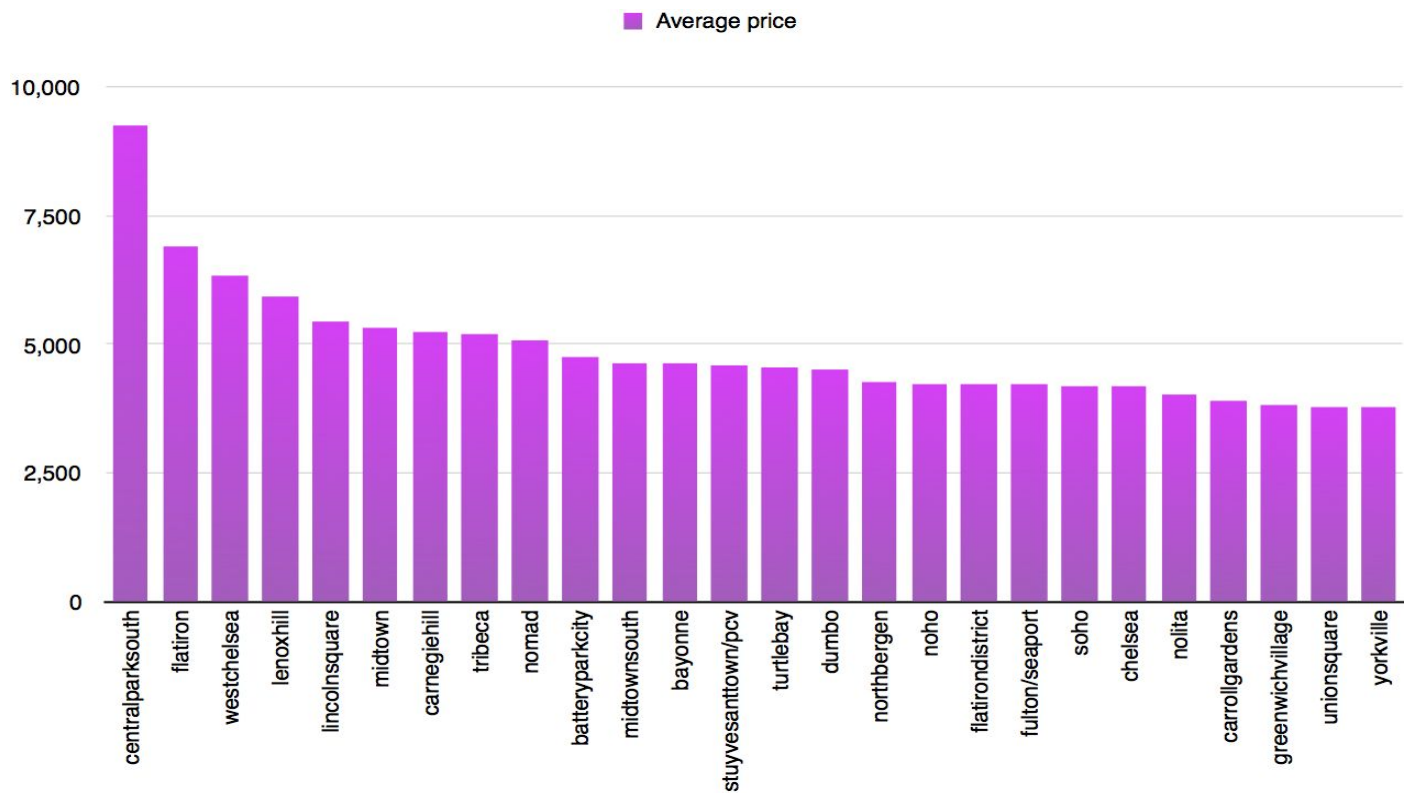
Aggregated Neighborhood Statistics

Most of our data came from listings for the Upper-East Side, Financial District, and Upper-West Side. This was to be expected since these are three of the largest neighborhoods in Manhattan by area so our results are consistent with our predicted outcome.

The highest average prices belonged to Central Park South, Flatiron, and West Chelsea respectively. These results were mostly consistent with our expectations, although we expected Tribeca and SoHo to rank higher and did not expect Central Park South to be such an extreme high-outlier.



Number of listings in the dataset that have more than 500 listings. The top five are: Upper East Side, Financial District, Upper West Side, Hell's Kitchen and East Village.

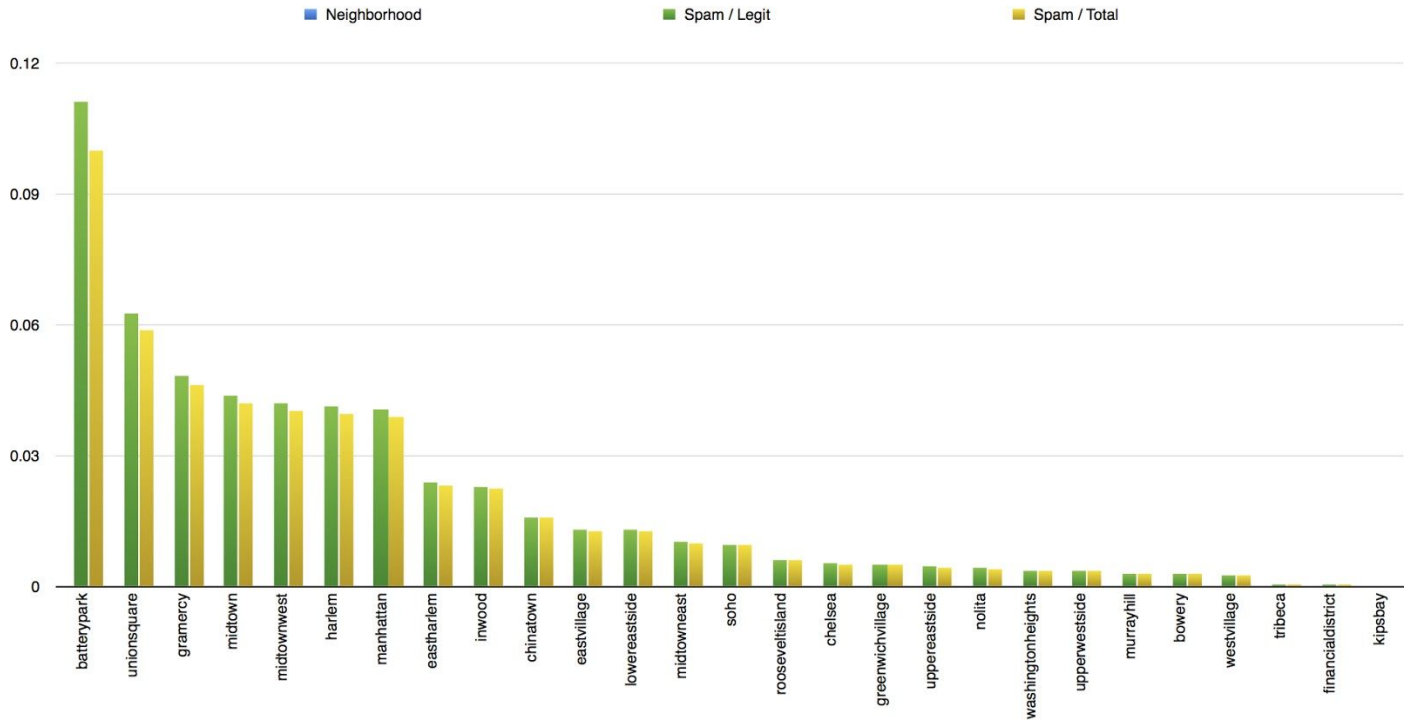


Top 30 neighborhoods that have the highest average rent. Top five results are Central Park South, Flatiron, West Chelsea, Lenox Hill and Lincoln Square. Each neighborhood has at least 10 listings in the data set.

Neighborhood Insights

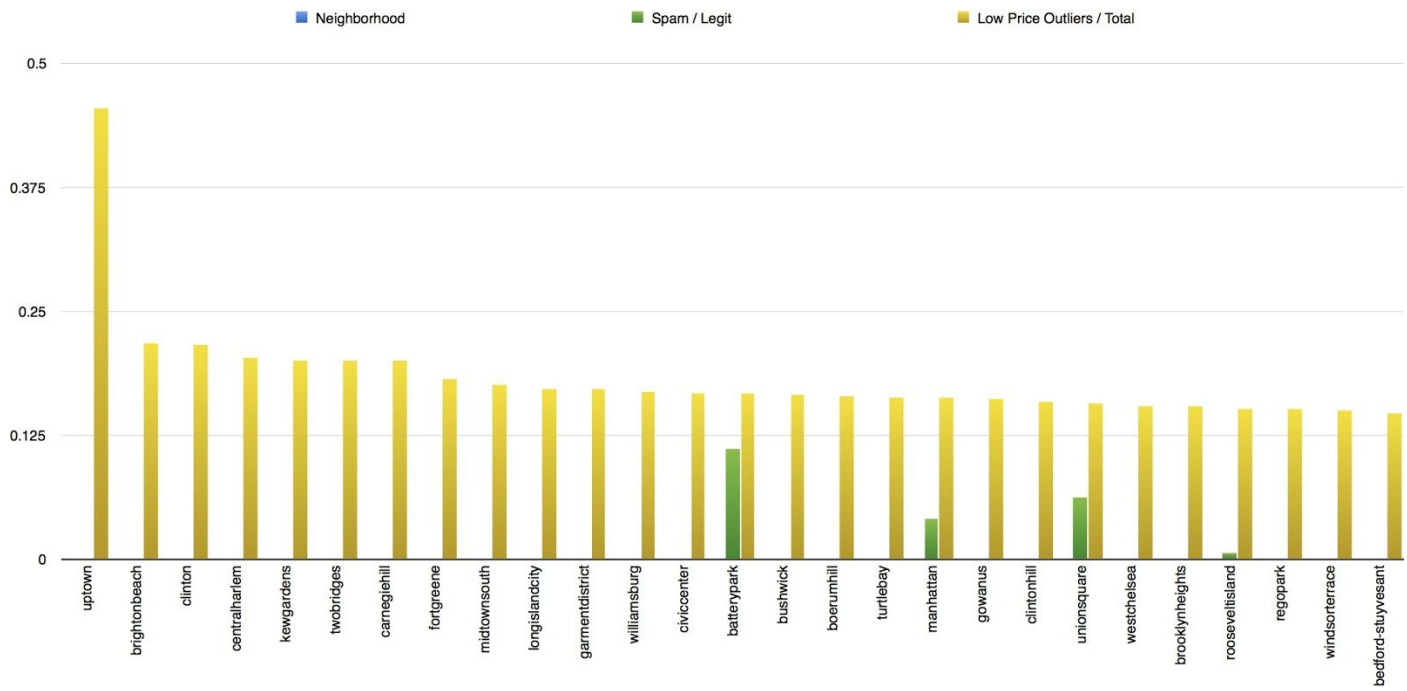
From the sample apartment listings that we extracted from our dataset, we identified price outliers by looking for apartment listings that are above or below 1 standard deviations of the mean price for that neighborhood (after removing spammy listings and listings with incomplete/unusable data). We classify "good value" listings as ones that have passed the spam filter and are at least 1 standard deviation below the mean (z-score <= -1)

When processing listings for a neighborhood, we also keep track of spam listings that are identified and output these counts for each neighborhood category. Neighborhoods with a high number of spam listings also tend to have a high number of good-value listings (since both tend to be low price outliers). For all neighborhoods we looked at, we found the following proportions for spam listings and spam/legit listings:



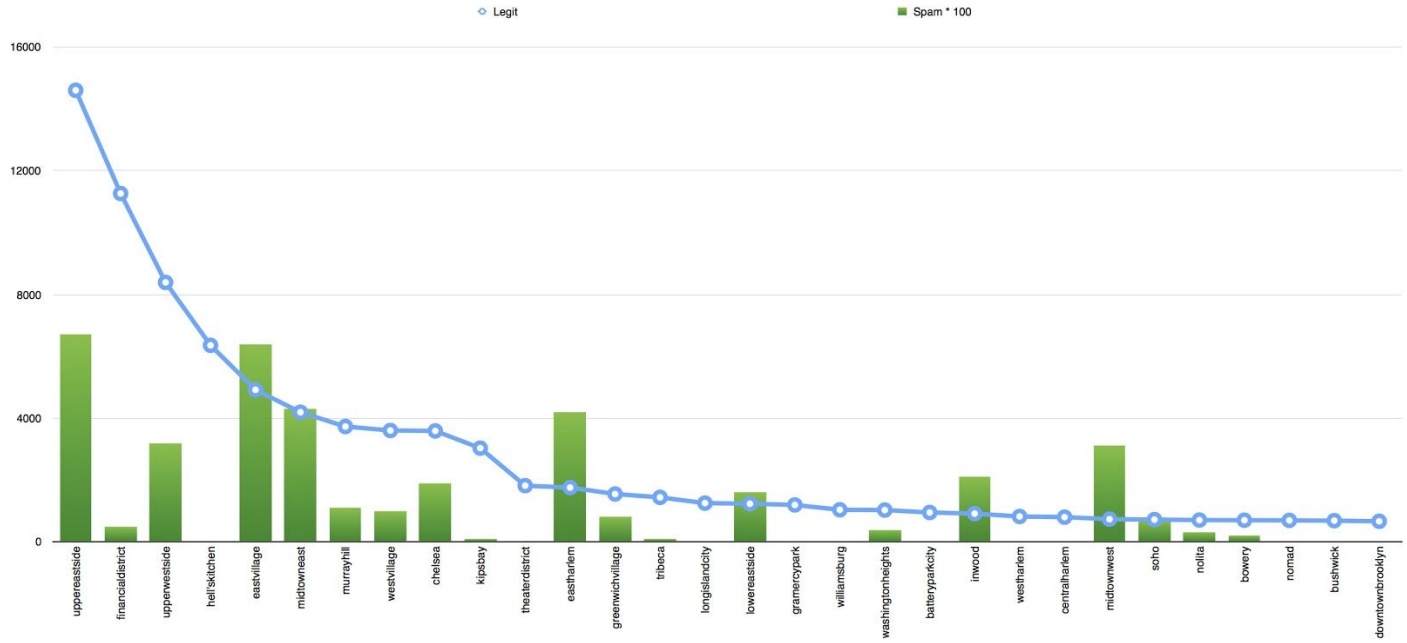
We found that Battery Park had the highest proportion of spammy listings of all the neighborhoods in our dataset. Upon further investigation, we found that neighborhoods with less well-defined boundaries such as Battery Park tend to have a higher percentage of listings that get marked as spam by our filter. For listings that exist in neighborhoods bordering areas that have lower price averages (or in areas with a larger standard deviation of prices like East Village), real-estate brokers will tend to label the apartment as being in the higher-priced neighborhood. While this is not technically fraud, these listings could be considered spam due to false advertising by the renter. In the case of Battery Park, it borders the Financial District, which is one of the highest priced neighborhoods in Manhattan.

On the other end of the spectrum, there are neighborhoods with a high ratio of low price outliers to total listings (shown in the chart below). Amongst these neighborhoods, Uptown, Clinton Hill, and Brighton Beach also had a low ratios of Spam/Legit listings, making them good areas to find good value apartments while avoiding scams.



Since some neighborhoods in our data set tend to be more represented than others, we also looked at the absolute number of spam listings for neighborhoods in our dataset with the most number of listings. The absolute listing counts for spam listings (for the neighborhoods with the largest counts) are as follows (total number of legit listings represented by the blue line):

(number of spam listings is multiplied by 100 for visibility)



The neighborhoods with the highest number of spam listings here are Upper East Side and East Village. Financial District has a surprisingly few number of spam listings despite having the most number of total listings, making it a great neighborhood for apartment seekers to look for apartments in.

VI. Future Work

Additional Computation:

This project could be expanded to consider price changes over a region. Currently, we classify listings as good value by looking at those that pass our spam filter and are low price-outliers. However, "good-value" listings could also be properties that are in high-growth or high-demand areas. To identify these cases, we could reference U.S. census data to track population changes in different areas and factor this into our calculation rather than simply looking for price outliers.

Data Collection Optimizations:

For a more large-scale implementation of this project, the crawler could be run in a distributed fashion on several nodes in a cluster. Running our data-collection job in a cluster could speed up the data-collection process and allow us to work with a larger data-set (although de-duplication and joining output files would be necessary in such a system).

Other Applications:

In the future, our project could be used to analyze the real-estate market for other population-dense cities. This project could also be applied to other online marketplace sites such as eBay or Facebook. We could follow a similar procedure to determine price outliers after flagging spammy listings, but the price distributions for other goods might be more unpredictable.

VII. Conclusion

In our analysis of New York City apartment listings, we uncovered some interesting neighborhood characteristics. We found that the Upper East Side had the highest number of listings across all three of our data sources (Craigslist, StreetEasy, Renthop) and Central Park South had the highest average price. For the most part, these results agreed with our preconceptions of New York City neighborhoods.

According to our spam and value computations, we found that Battery Park, East Village, and Upper East Side were neighborhoods with relatively high amounts of spam. The Financial District, Uptown, Clinton Hill, and Brighton Beach were neighborhoods with a high number of good values and low amount of spam, making them the ideal neighborhoods to search for apartments in.

Using the spam evaluation metric that we developed, we were able to uncover spam listings with reasonable accuracy. From the list of 497 spam listings that our filter produced, we evaluated a random sample of 50 listings across different neighborhoods. We compared the results for these 50 listings against a consensus evaluation of three human judges and found that the filter accurately identified 33 of these spam listings while 17 were false-positives -- an accuracy rate of 66%.

ACKNOWLEDGMENT

We'd like to thank our Professor Suzanne McIntosh for advising us on this project and our T.A. Hassan Zaidi for answering our questions related to this project. We'd also like to thank Craigslist, Renthop, and StreetEasy for supplying the data that we used in our analysis.

REFERENCES

[1] T. White. Hadoop: The Definitive Guide. O'Reilly Media Inc., Sebastopol, CA, May 2012.

[2] A. Gates. Programming Pig. O'Reilly Media Inc.,Sebastopol, CA, October 2011.

[3] J. Dean and S. Ghemawat. MapReduce: Simplified data processing on large clusters. In proceedings of 6th Symposium on Operating Systems Design and Implementation, 2004.

[4] C. Olston, B. Reed, U. Srivastava, R. Kumar, A. Tomkins. Pig Latin: A not-so-foreign language for data processing. In proceedings of SIGMOD, June 2008.

[5] Curtis A. Smith," Detecting anomalies in Data using Benford's law"

[6] A. Arasu, and H. Garcia-Molina. Extracting structured data from Web pages. Proc. ACM SIGMOD Conf., 2003.

[7] A. Katal, "Big Data: Issues, Challenges, Tools and Good Practices", The Sixth International Conference on Contemporary Computing. pp. 404-409, 2013.

[8] A. Fahad, "A survey of clustering algorithms for big data: Taxonomy and empirical analysis", *IEEE Trans. Emerg. Topics Comput.*, vol. 2, no. 3, pp. 267-279, 2014

[9] Sasaki M, Shinnou H (2005) Spam detection using text clustering. In: Proceedings of international conference on cyberworlds, CW2005.

[10] M. Thelwall, "A web crawler design for data mining," J. Journal of Information Science, vol. 27