DSO 562: Fraud Analytics

Unsupervised Fraud Model on the NY data

Trojan Consulting Team 3

Ruozhang (Olivia) Yao

Table of Content

Executive Summary

Objective:

Real estate fraud is when one person or party makes misrepresentation or uses false information to take advantage of the other party during a real estate sale/purchase. Real estate fraud can cause enormous losses especially for government like New York City. This report examined the New York property data to find potential fraud with methods include Principal Component Analysis (PCA), Heuristic Algorithm and Autoencoder. Data is processed and analyzed in Python and R.

Project Outline:

The original dataset consists of information including sizes, building classes, values, tax classes owner of about 1 million New York properties. The general process of analysis step includes:

1. Data cleaning and missing data filing. We proposed the dataset to optimize the results of the analysis. we filed missing values by methods like using mean of the group the missing value belongs to or mode of properties which close to it in geographical location.
2. Building expert variables and standardizing. we built special variables that look for fraud model and scaled them before put them into machine learning models.
3. Dimensionality reduction through the PCA process. We only keep the main PCs and Z scale the data field again.
4. Applying fraud algorithm, calculating fraud score identifying potential fraud. We combined the Z scores as the first fraud score and used the reconstruction error in the process of autoencoder as the second score. Final fraud score is a combination of these two scores.

With the highest Cumulative average of Heuristic Fraud scores and Autoencoder Fraud scores, we found the top 10 records which look anomalous and could be classified as underlying real estate frauds.

Detailed examination on top 10 high scores shows that abnormalities mainly due to three reasons:

1. Accidentally input wrong data
2. Falsely report property value to cheat banks
3. Tax avoidance

Part I. Data Description

Data Summary

The Property Valuation and Assessment Data represents NYC properties assessments for the purpose of calculating Property Tax, Grant Eligible Properties Exemptions and/or Abatements. Data was collected and entered into the system by various City employees, like Property Assessors, Property Exemption specialists, ACRIS reporting, and Department of Building reporting.

The data was created on September 2, 2011 and updated annually. The final assessment time of this dataset is 2011-11-01. Therefore, it covers various information of each property in New York City by November 1, 2011. There are 1070994 records and 32 fields in the dataset.

Following is description of the variables we consider to be the most important. The complete Data Quality Report can be found in appendix.

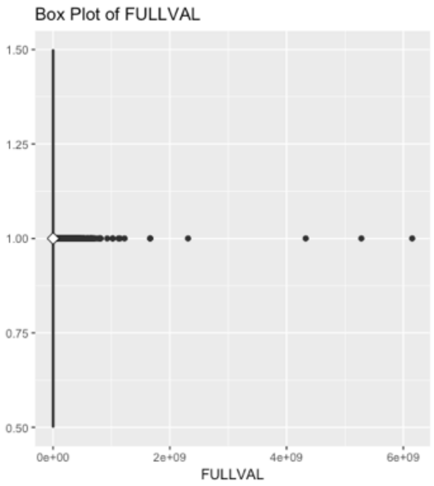
Key Variables Description

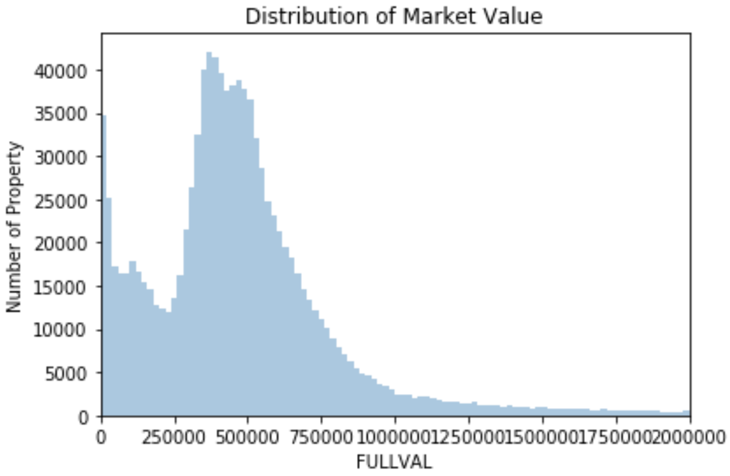
Field Name: FULLVAL (numeric, dtype: float64)

Description:
Market value of the property in dollars. There is no missing data while there are 13007 properties (1.21%) with the market value of 0 in the dataset. The statistics excluding FULL VAL is 0 are shown as below.

```
count    1.057987e+06
mean     8.850128e+05
std      1.165300e+07
min      4.000000e+00
25%      3.110000e+05
50%      4.500000e+05
75%      6.230000e+05
max      6.150000e+09
```

Unique Value:
From the boxplot, we can see that number of properties declines exponentially with increase of market value, and there are outliers in FULLVAL. Therefore, we drew a log distribution of FULLVAL.



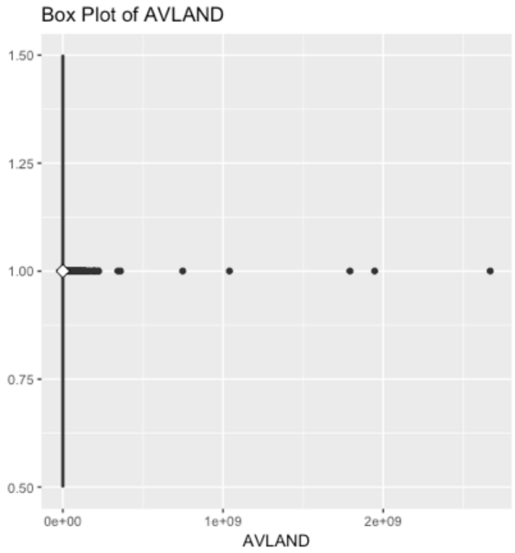


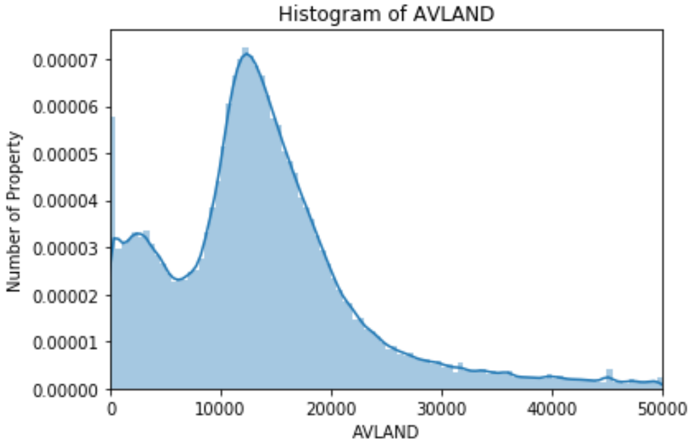
Field Name: AVLAND (numeric, dtype: float64)

Description:
Actual land value of the property in dollars. There is no missing data while there are 13009 properties with the AVLAND of 0 in the dataset. The statistics excluding AVLAND is 0 are shown as below.

```
count    1.057985e+06
mean     8.611392e+04
std      4.082117e+06
min      1.000000e+00
25%      9.445000e+03
50%      1.378200e+04
75%      1.986000e+04
max      2.668500e+09
```

Unique Value:
From the boxplot, we can see are outliers in AVLAND and 75% of properties AVLAND are less than 20000. The distribution with AVLAND < 50,000 are shown as below.





Field Name: ZIP (categorical, 5 digit)

Description:
196 unique Zip codes. There is missing data such that only 1041104 properties have non-null ZIP. There are 196 unique zip code values. (NA = 29890)

Unique Value:
There are 24606 properties in zip code 10314.

Part II. Data Cleaning

Filling ZIP

1. For NULLs or 0's, group by BLOCK, calculate and fill with mode value of group (Find the most frequently shown Zip code among a block to fill the NA)
2. For BLOCK in which all the ZIP are missing, run a function(zoo::na.locf) to return the nearest non-NA value to the rest of NAs (eg: FOR BLOCK 2502, if all ZIP in BLOCK 2501 AND 2503 are 11378, fill the BLOCK 2502 with 11378, too)

Filling FULLVAL, AVLAND, AVTOT, LTFRONT, LTDEPTH, BLDFRONT and BLDDEPTH

1. For NULLs or 0's, group by BLDGCL and new ZIP, calculate and fill with median value of group if size of group is more than 5;
2. For remaining NULLs or 0's, group by TAXCLASS and new ZIP, calculate and fill with median value of group;
3. For remaining NULLs or 0's, group by TAXCLASS and B, calculate and fill with median value of group;
4. For remaining NULLs or 0's, group by TAXCLASS, calculate and fill with median value of group

We followed a consistent replacement method for the following seven variables that needed to be filled: FULLVAL, AVLAND, AVTOT, LTFRONT, LTDEPTH, BLDFRONT and BLDDEPTH. Our process contained four steps, first aggregating by combinations of location variable and a tax related variables, then filling the remaining variables with the median value of the groups.

Our first aggregation was our most specific, using BLDGCL or building tax class and our new ZIP variable. After grouping all properties by these values and filtering out blank rows, we created two new columns, one with the median value of the current variable for the grouping and one with the count of properties within that grouping. We then joined this newly created data frame to our full data using a left-join by both BLDGCL and our new ZIP variable.

Next, we followed an identical procedure and created a data frame of counts and median values broadened our aggregation by using TAXCLASS and our new ZIP values rather than BLDGCL. We performed a similar left-join and added the frame to our dataset.

Third, we again aggregated by a tax related variable and a property related variable, but once again broadened our variable selection to group by B or borough and TAXCLASS and left-joined similarly.

Finally, we filtered and grouped our data by TAXCLASS, since the variable had no missing values. This served as our fill backstop should the rest of our aggregated fields fail to fill the missing fields.

After joining all of our newly aggregated fields to our data frame, we began our filling process. For our first aggregation, given its specificity, we filled our FULLVAL column of all NULL or 0 values if the count for the group was larger than 5.

Following our first fill, we continued to our second, third and finally fourth fill, updating all 0 or NULL values with the associated value of the new columns starting with the most specific and becoming broader until all remaining cells were filled by our backstop, only TAXCLASS grouping.

Lastly, we removed the new columns we created and continued on to the next variable of the seven that would use the same fill process until all NULL and 0 values were filled for FULLVAL, AVLAND, AVTOT, LTFRONT, LTDEPTH, BLDFRONT and DBLDEPTH.

Filling STORIES

1. For NULLs or 0's, group by BLOCK and TAXCLASS, calculate and fill with median value of group
2. For remaining NULLs or 0's, group by TAXCLASS and new ZIP, calculate and fill with median value of group
3. For remaining NULLs or 0's, group by TAXCLASS and B, calculate and fill with median value of group
4. For remaining NULLs or 0's, group by TAXCLASS, calculate and fill with median value of group

The last field that contained missing values was the STORIES field. For stories, our general process was very similar, but our specific variable choice at each level of aggregation was different. Due to the presence of large, outlier, skyscraper buildings, we began with an even more specific aggregation of location. We group by BLOCK and TAXCLASS to create our first layer, followed by combinations of ZIP and TAXCLASS, B and TAXCLASS and finally simply TAXCLASS as our backstop.

Part III. Variable Creation

The variables that we decided to create all came down to different transformations of three important variables. These variables were the following:

V_1 = **FULLVAL** – the full value of the building

V_2 = **AVLAND** – the assessed value of land

V_3 = **AVTOT** – the assessed value of property

Then, we created three new variables that we would use to transform the original variables to create 9 key ratios that we would use to detect fraud. The following are the new variables we created from original variables, to be used later on to create the key ratios:

V_4 = **LTFRONT * LTDEPTH** – the area of the lot

V_5 = **BLDFRONT * BLDDEPTH** – the area of the building

V_6 = V_5 *** STORIES** – the volume of the building

The next step was to actually create the key ratio variables. V_1 , V_2 , and V_3 all served as denominators in our key ratios, while V_4 , V_5 , and V_6 all served as the denominators. We created nine ratios in total, using all combinations of the six.

- $R_1 = V_1/V_4$
- $R_2 = V_1/V_5$
- $R_3 = V_1/V_6$
- $R_4 = V_2/V_4$
- $R_5 = V_2/V_5$
- $R_6 = V_2/V_6$
- $R_7 = V_3/V_4$
- $R_8 = V_3/V_5$
- $R_9 = V_3/V_6$

Next, we want to group each of these variables by variables that would make outliers more noticeable. For example, we want to group the variables in a way such that a low property value in a poor neighborhood of Queens would not score highly in our model, because it is being compared against similar property, not all property of New York. For that reason, we decided on five different criteria that we would group by and obtain the mean, then divide the all of the nine ratios by. The following were the five criteria we would group by:

- **ZIP** – zip code
- **ZIP3** – 3-digit zip code
- **TAXCLASS** – the tax class of the building
- **B** – borough of the building
- **ALL** – grouped by **ZIP**, **TAXCLASS**, and **B**. Note that **ZIP** is a more specific version of **ZIP3**, so it is redundant to group by that as well.

After grouping the data and calculating the means for each combination of these, we created 45 variables by dividing the 9 ratios of each record by the mean ratio value of its group. The following is another representation of 45 ratios created:

$$\frac{R_1}{\overline{(R_1)_g}}, \frac{R_2}{\overline{(R_2)_g}}, \frac{R_3}{\overline{(R_3)_g}} \dots \frac{R_9}{\overline{(R_9)_g}}\ Where\ g\ is\ each\ of\ the\ five\ groups$$

After calculating these ratios for each of the 45 combinations, we ended up with 45 variables that we would use to create the fraud score for each record. The reasoning behind the creation of these variables is that given a buildings location and building type, we shouldn’t expect much deviation in terms of the ratio of the value of a property versus the area of the property. So, after the creation of these 45 variables, we moved forward to dimensionality reduction.

Part IV. Dimensionality Reduction

After creating expert variables, we start the process of standardizing those variables and reducing the dimensionality using Principal Component Analysis(PCA).

We first used python to standardiz each variable by subtracting the mean of the variable from each record and dividing them by the standard deviation of this variable. This process leave us with 45 columns of z-scores. We then use this z-score for PCA.

We then imported the decomposition package from scikit-learn and specified the number of principal scores we need to be 10 by using “n_components=10”. We use 10 principal component scores because we want to catch as much variation as possible with least number of principal scores as possible. So we tend to keep the least number of scores possible to explain around 80 to 90% of the total variance. With 10 principal component scores, we can explain around 90% of the variation of the entire dataset. We then discard the higher PCs and we reduce our dimensions from n to just a few.

After calculating principal component analysis, we have 10 columns of 10 principal scores. We again standardize those scores with the z-scaling methods we used before. After reducing dimensionality, we are ready to build our fraud scores using different algorithms.

Part V. Algorithms

We chose the Heuristic Function of z-scores and an Auto encoder as our Fraud score algorithms to get two scores: Score1 and Score2 respectively. After scoring all the records, we did the following steps:

a) Sort records by 'Score1' in ascending order
b) Replace 'Score1' with the sorted rank order (Lowest score getting rank 1 and so on)
c) Sort records by 'Score2' in ascending order
d) Replace 'Score2' with the sorted rank order (Lowest score getting rank 1 and so on)
e) Score1 and Score2 are on same scale now and we took their average to get 'Cumulative score', which is our Final Fraud score.

Method 1 - Heuristic Function of z-scores (Score1)

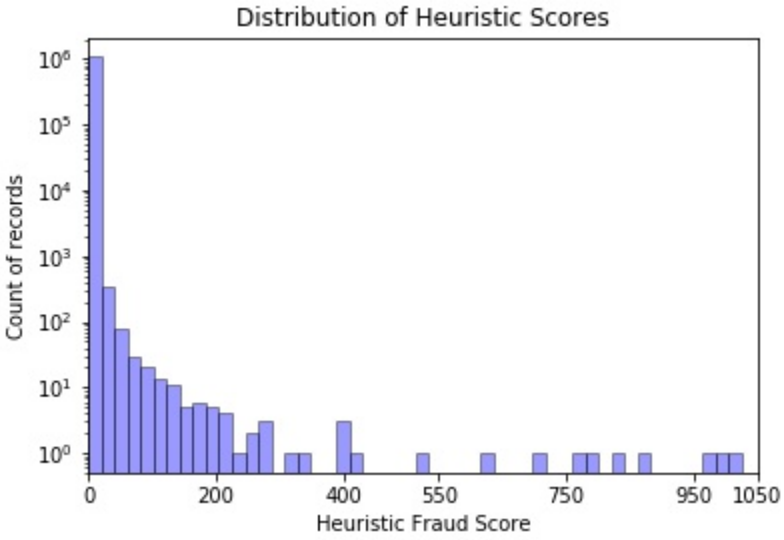
After we are done with PCA for reducing dimensionality and we go ahead with 10 Principal components (PCs), the value of each variable for a record explicitly shows how unusual that record is. These variables are z-scaled and the values that we try to compare are on the same scale and are known as z-scores.

We'll use Euclidean distance to compute distance between two points (essentially z-scores) in our data space. The Fraud score is a second-degree function of these z-scores that look for extremes in our records. We keep n=2 to ensure we compute Euclidean distance between any two points in our data space. The following is the formula for computing the Euclidean distance in terms of z-scores obtained from the last section of PCA:

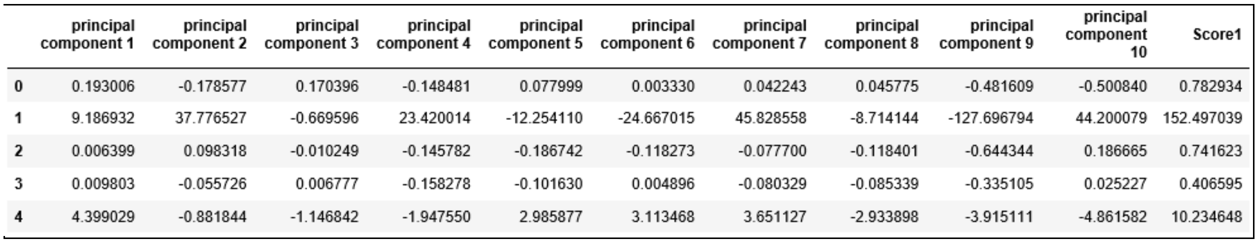
$$s_i = \left(\sum_k |z_k^i|^n \right)^{1/n}$$

, where n = 2 and k goes from 1 to 10

Following Histogram shows the distribution of our Fraud scores based on Heuristic Function of z-scores



Snippet of our Score1 based on Euclidean Distance (Heuristic Function of z-scores)



	principal component 1	principal component 2	principal component 3	principal component 4	principal component 5	principal component 6	principal component 7	principal component 8	principal component 9	principal component 10	Score1
0	0.193006	-0.178577	0.170396	-0.148481	0.077999	0.003330	0.042243	0.045775	-0.481609	-0.500840	0.782934
1	9.186932	37.776527	-0.669596	23.420014	-12.254110	-24.667015	45.828558	-8.714144	-127.696794	44.200079	152.497039
2	0.006399	0.098318	-0.010249	-0.145782	-0.186742	-0.118273	-0.077700	-0.118401	-0.644344	0.186665	0.741623
3	0.009803	-0.055726	0.006777	-0.158278	-0.101630	0.004896	-0.080329	-0.085339	-0.335105	0.025227	0.406595
4	4.399029	-0.881844	-1.146842	-1.947550	2.985877	3.113468	3.651127	-2.933898	-3.915111	-4.861582	10.234648

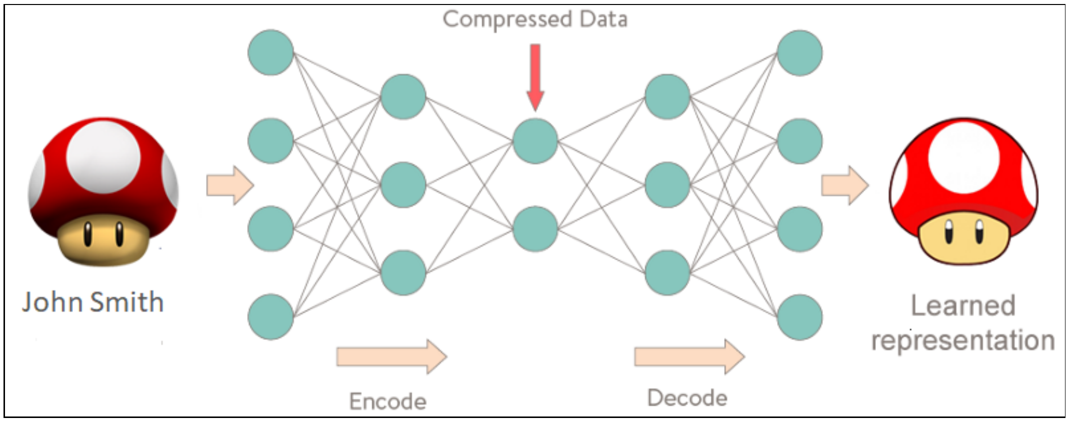
\

Method 2 - Use Autoencoder to compute Fraud scores (Score2)

After we finished using PCA to reduce dimensionality and we continued with 10 Principal components (PCs), the value of each variable for a record explicitly shows how unusual that record is. These variables are z-scaled and the values that we try to compare are on same scale and are known as z- scores. We will discuss the Autoencoder in this section and how it can be an effective tool to compute Fraud scores.

Autoencoder is an unsupervised learning technique that aims at regenerating inputs supplied to the Autoencoder Neural network with minimal possible errors, but if there are certain inputs (record values) that are quite far from the general representation of the data set, autoencoder model will present an error in reproducing them. We call this error as Reproduction Error and we will treat these record values as possible anomalies in our dataset.

Visual Representation of an Autoencoder in simple language



Note: We haven't used Autoencoder for dimensionality reduction (PCA has been used)

Now we will follow the following steps to train our autoencoder on the entire data set and compute the reproduction error for each record thus generating our Fraud Score2.

a) We start with installing relevant packages: **TensorFlow and Keras** in Python environment and then we run our autoencoder model on the z-scaled 10 PC record values. We use Rectified Linear Units (**ReLu**) as an activation function for our hidden layers (Compressed Data layer in the figure above) which looks at interaction between different variables and non-linear effect for different values of a variable while predicting the output (in our case regenerating the whole data set).

b) For our output layer (which is the last layer in the whole model), we used hyperbolic tangent (**tanh**) function as an activation function.

c) After this we predicted values of input layers using our trained autoencoder model on the whole data set and got predicted autoencoder values.

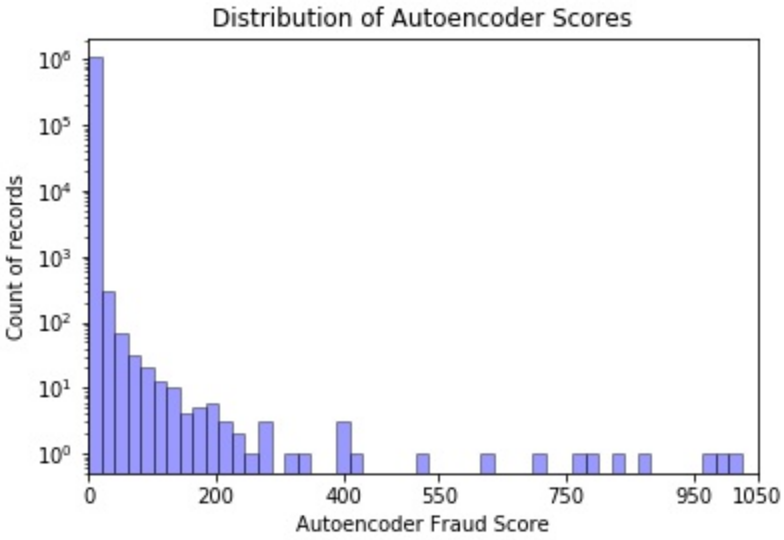
d) Further we compare these predicted z-scaled values to the original z-scaled values that we had after doing PCA for each record value and will compute the reproduction error. Reproduction error is computed as below:

$$\boxed{s_i = \left(\sum_k |z_k^{\prime i} - z_k^i|^n \right)^{1/n}}$$

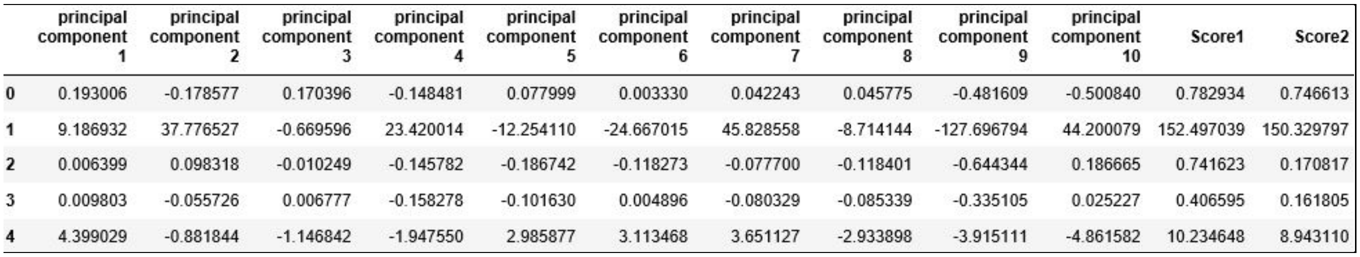
, where n =2 and k goes from 1 …10

e) A measure of the reproduction error is a measure of unusualness of a record and is thus a fraud score.

Following Histogram shows the distribution of our Fraud scores based on Autoencoder and Reproduction error model



Snippet of our Score2 based on Autoencoder and reproduction error



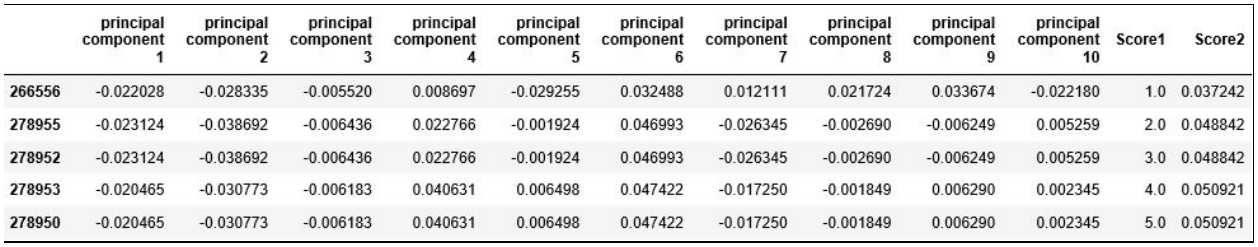
	principal component 1	principal component 2	principal component 3	principal component 4	principal component 5	principal component 6	principal component 7	principal component 8	principal component 9	principal component 10	Score1	Score2
0	0.193006	-0.178577	0.170396	-0.148481	0.077999	0.003330	0.042243	0.045775	-0.481609	-0.500840	0.782934	0.746613
1	9.186932	37.776527	-0.669596	23.420014	-12.254110	-24.667015	45.828558	-8.714144	-127.696794	44.200079	152.497039	150.329797
2	0.006399	0.098318	-0.010249	-0.145782	-0.186742	-0.118273	-0.077700	-0.118401	-0.644344	0.186665	0.741623	0.170817
3	0.009803	-0.055726	0.006777	-0.158278	-0.101630	0.004896	-0.080329	-0.085339	-0.335105	0.025227	0.406595	0.161805
4	4.399029	-0.881844	-1.146842	-1.947550	2.985877	3.113468	3.651127	-2.933898	-3.915111	-4.861582	10.234648	8.943110

Calculating Cumulative Average Scores-

After scoring all the records, we did the following steps:

a) Sort records by 'Score1' in ascending order
b) Replace 'Score1' with the sorted rank order (Lowest score getting rank 1 and so on)

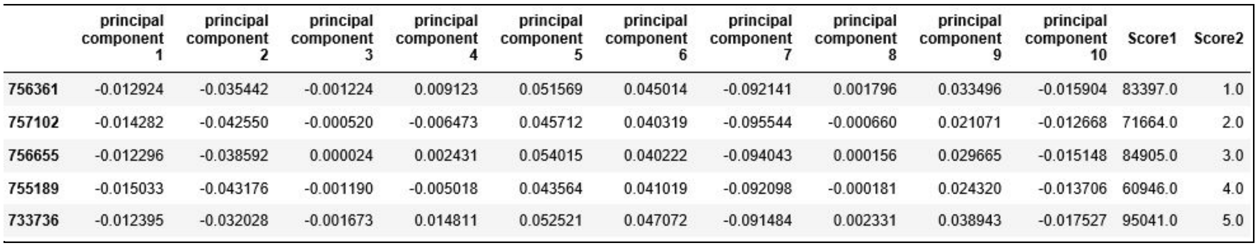
Snippet of our dataset up to the above-mentioned steps a and b:



	principal component 1	principal component 2	principal component 3	principal component 4	principal component 5	principal component 6	principal component 7	principal component 8	principal component 9	principal component 10	Score1	Score2
266556	-0.022028	-0.028335	-0.005520	0.008697	-0.029255	0.032488	0.012111	0.021724	0.033674	-0.022180	1.0	0.037242
278955	-0.023124	-0.038692	-0.006436	0.022766	-0.001924	0.046993	-0.026345	-0.002690	-0.006249	0.005259	2.0	0.048842
278952	-0.023124	-0.038692	-0.006436	0.022766	-0.001924	0.046993	-0.026345	-0.002690	-0.006249	0.005259	3.0	0.048842
278953	-0.020465	-0.030773	-0.006183	0.040631	0.006498	0.047422	-0.017250	-0.001849	0.006290	0.002345	4.0	0.050921
278950	-0.020465	-0.030773	-0.006183	0.040631	0.006498	0.047422	-0.017250	-0.001849	0.006290	0.002345	5.0	0.050921

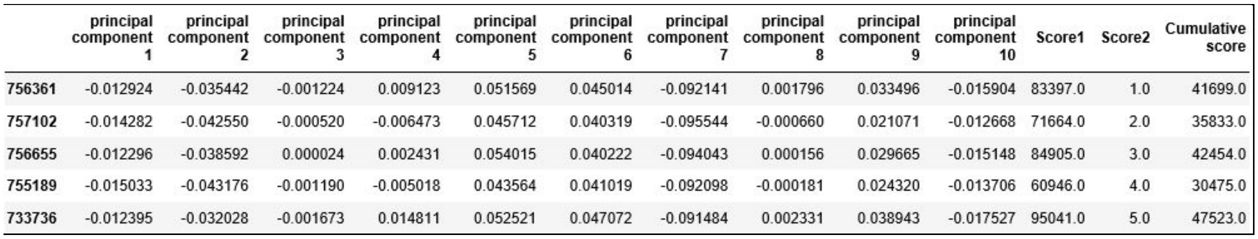
c) Sort records by 'Score2' in ascending order
d) Replace 'Score2' with the sorted rank order (Lowest score getting rank 1 and so on)

Snippet of our dataset up to the above-mentioned steps c and d:



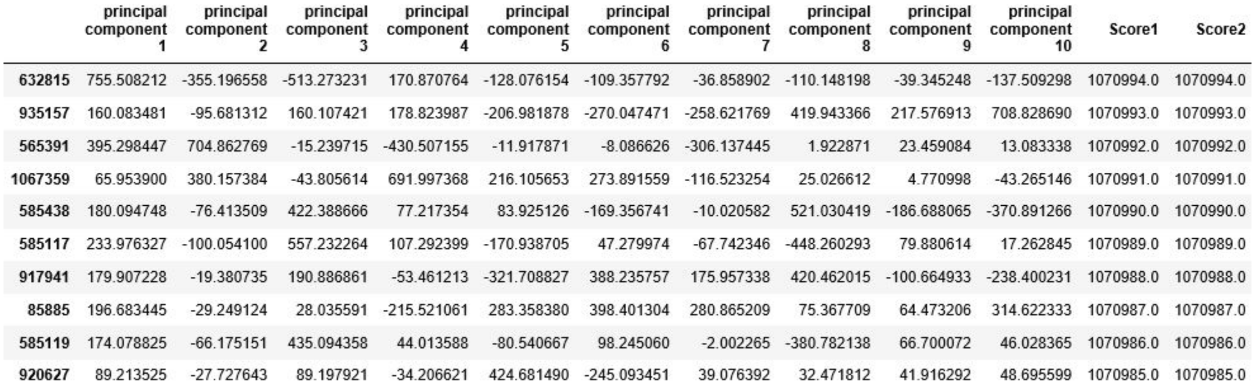
	principal component 1	principal component 2	principal component 3	principal component 4	principal component 5	principal component 6	principal component 7	principal component 8	principal component 9	principal component 10	Score1	Score2
756361	-0.012924	-0.035442	-0.001224	0.009123	0.051569	0.045014	-0.092141	0.001796	0.033496	-0.015904	83397.0	1.0
757102	-0.014282	-0.042550	-0.000520	-0.006473	0.045712	0.040319	-0.095544	-0.000660	0.021071	-0.012668	71664.0	2.0
756655	-0.012296	-0.038592	0.000024	0.002431	0.054015	0.040222	-0.094043	0.000156	0.029665	-0.015148	84905.0	3.0
755189	-0.015033	-0.043176	-0.001190	-0.005018	0.043564	0.041019	-0.092098	-0.000181	0.024320	-0.013706	60946.0	4.0
733736	-0.012395	-0.032028	-0.001673	0.014811	0.052521	0.047072	-0.091484	0.002331	0.038943	-0.017527	95041.0	5.0

e) Score1 and Score2 are on same scale now and we took their average to get the 'Cumulative score', which is our Final Fraud score



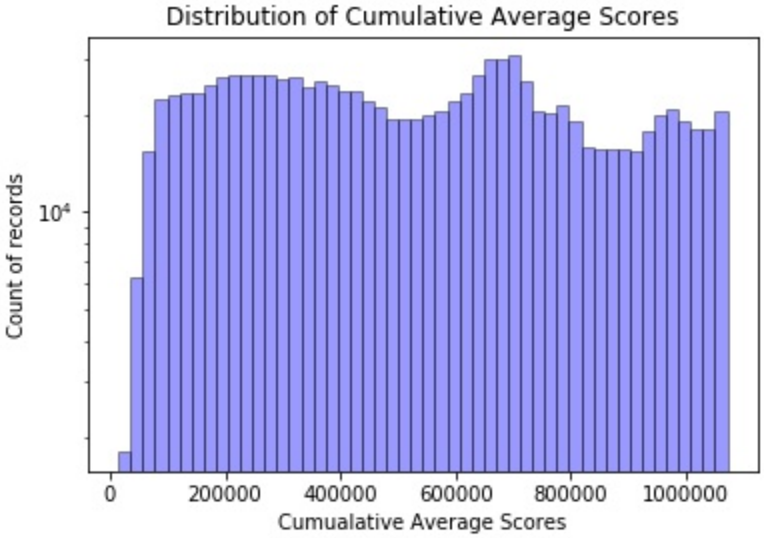
	principal component 1	principal component 2	principal component 3	principal component 4	principal component 5	principal component 6	principal component 7	principal component 8	principal component 9	principal component 10	Score1	Score2	Cumulative score
756361	-0.012924	-0.035442	-0.001224	0.009123	0.051569	0.045014	-0.092141	0.001796	0.033496	-0.015904	83397.0	1.0	41699.0
757102	-0.014282	-0.042550	-0.000520	-0.006473	0.045712	0.040319	-0.095544	-0.000660	0.021071	-0.012668	71664.0	2.0	35833.0
756655	-0.012296	-0.038592	0.000024	0.002431	0.054015	0.040222	-0.094043	0.000156	0.029665	-0.015148	84905.0	3.0	42454.0
755189	-0.015033	-0.043176	-0.001190	-0.005018	0.043564	0.041019	-0.092098	-0.000181	0.024320	-0.013706	60946.0	4.0	30475.0
733736	-0.012395	-0.032028	-0.001673	0.014811	0.052521	0.047072	-0.091484	0.002331	0.038943	-0.017527	95041.0	5.0	47523.0

f) Now, we look at the Top 10 Fraud scores sorted in descending order of 'Cumulative Score'



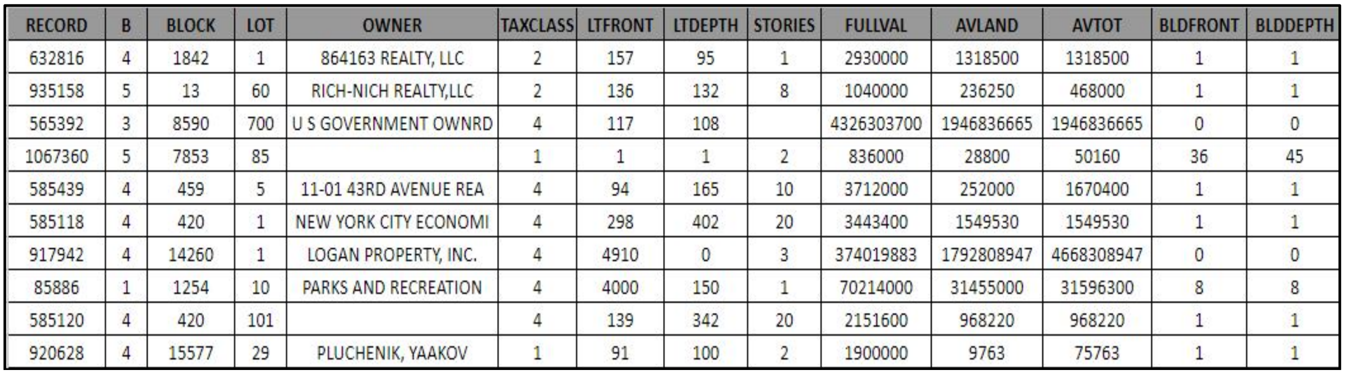
	principal component 1	principal component 2	principal component 3	principal component 4	principal component 5	principal component 6	principal component 7	principal component 8	principal component 9	principal component 10	Score1	Score2
632815	755.508212	-355.196558	-513.273231	170.870764	-128.076154	-109.357792	-36.858902	-110.148198	-39.345248	-137.509298	1070994.0	1070994.0
935157	160.083481	-95.681312	160.107421	178.823987	-206.981878	-270.047471	-258.621769	419.943366	217.576913	708.828690	1070993.0	1070993.0
565391	395.298447	704.862769	-15.239715	-430.507155	-11.917871	-8.086626	-306.137445	1.922871	23.459084	13.083338	1070992.0	1070992.0
1067359	65.953900	380.157384	-43.805614	691.997368	216.105653	273.891559	-116.523254	25.026612	4.770998	-43.265146	1070991.0	1070991.0
585438	180.094748	-76.413509	422.388666	77.217354	83.925126	-169.356741	-10.020582	521.030419	-186.688065	-370.891266	1070990.0	1070990.0
585117	233.976327	-100.054100	557.232264	107.292399	-170.938705	47.279974	-67.742346	-448.260293	79.880614	17.262845	1070989.0	1070989.0
917941	179.907228	-19.380735	190.886861	-53.461213	-321.708827	388.235757	175.957338	420.462015	-100.664933	-238.400231	1070988.0	1070988.0
85885	196.683445	-29.249124	28.035591	-215.521061	283.358380	398.401304	280.865209	75.367709	64.473206	314.622333	1070987.0	1070987.0
585119	174.078825	-66.175151	435.094358	44.013588	-80.540667	98.245060	-2.002265	-380.782138	66.700072	46.028365	1070986.0	1070986.0
920627	89.213525	-27.727643	89.197921	-34.206621	424.681490	-245.093451	39.076392	32.471812	41.916292	48.695599	1070985.0	1070985.0

Following Histogram shows the distribution of our cumulative average of Fraud scores



Part VI. Results

The Top 10 Records with the highest Cumulative average of Heuristic Fraud scores and Autoencoder Fraud scores:



RECORD	B	BLOCK	LOT	OWNER	TAXCLASS	LTFRONT	LTDEPTH	STORIES	FULLVAL	AVLAND	AVTOT	BLDFRONT	BLDDEPTH
632816	4	1842	1	864163 REALTY, LLC	2	157	95	1	2930000	1318500	1318500	1	1
935158	5	13	60	RICH-NICH REALTY,LLC	2	136	132	8	1040000	236250	468000	1	1
565392	3	8590	700	U S GOVERNMENT OWNRD	4	117	108		4326303700	1946836665	1946836665	0	0
1067360	5	7853	85		1	1	1	2	836000	28800	50160	36	45
585439	4	459	5	11-01 43RD AVENUE REA	4	94	165	10	3712000	252000	1670400	1	1
585118	4	420	1	NEW YORK CITY ECONOMI	4	298	402	20	3443400	1549530	1549530	1	1
917942	4	14260	1	LOGAN PROPERTY, INC.	4	4910	0	3	374019883	1792808947	4668308947	0	0
85886	1	1254	10	PARKS AND RECREATION	4	4000	150	1	70214000	31455000	31596300	8	8
585120	4	420	101		4	139	342	20	2151600	968220	968220	1	1
920628	4	15577	29	PLUCHENIK, YAAKOV	1	91	100	2	1900000	9763	75763	1	1

Our methodology to work with the top 10 records that have high fraudulent scores was to look at the values of important independent variables (B, BLOCK, LOT, TAXCLASS, LRFRONT, LTDEPTH, STORIES, FULLVAL, AVLAND, AVTOT, BLDFRONT, BLDDEPTH) straight in our original NY_Property data set and try to identify possible reasons of fraudulent behavior.

Let's quickly revisit the values of 3 important variables that we felt were very important in deciding possible reasons of a fraudulent behavior for a property

a) FULLVAL - Market Value of a property (done by a valuation company)
b) AVLAND - Actual land value of a property (not including the construction)
c) AVTOT - Actual total value of a property/ land/ other structure (including all constructions)

Detailed Analysis of Top 10 records

a) **632816 –** Looking at the property details (TAXCLASS) and owner, we can say that it is a commercial building and would probably house apartments within it, but very low values of BLDFRONT and BLDDEPTH look susceptible and open to investigation. There might be a case that this apartment is under construction, but an investigation is required to identify if this is a possible case of forging property characteristics to avoid less payment of taxes.

b) **935158** - This record number contains information of an 8-story building with its LTFRONT and LTDEPTH value higher than top 75% of the corresponding values in our dataset and looking at property characteristics and its neighborhood, it's hard to believe that the property will be valued around $ 0.46 M (AVTOT). This can be a possible case of misleading property values to avoid higher taxes.

c) **565392** - Given the available land area, property has a very high valuation of property ($4 B). This is evident from the values of LOTFRONT and LODEPTH, which are around 100 feet. This particular record value should definitely indicate a red flag among all records and should be scrutinized further to check if there is a possible fraudulent situation.

d) **1067360** - This record can be a possible fraudulent property record considering the huge difference between AVTOT and FULLVAL. FULL VAL is around 16 times higher than

AVTOT. Also, AVTOT is low and the property's dimensions don't make much sense either when we look at the high values of BLDFRONT and BLDDEPTH when compared to LTFRONT and LTDEPTH.

e) **585439** - This record corresponds to a hotel in New York and looking at the 10-story building, it's difficult to believe that it will value around $ 3.7 M (FULLVAL). Also, there is not much construction on the land (since BLDFRONT and BLDDEPTH are 1 foot each), but still AVTOT is around 16 times higher than AVLAND, so this can be a possible situation of misleading property values to avoid higher taxes.

f) **585118**- This record corresponds to a 20-story building in NY with quite higher values of LTFRONT and LTDEPTH (298 ft and 402 ft respectively) and with very low property values. FULLVAL is around $3.5M which doesn't justify the characteristics of the property. So, this is a possible case of fraud by using framed AVTOT (actual total value of a property) and FULLVAL (total market value of a property). Also, values of BLDFRONT and BLDDPETH as 1 ft each, which don't match with this property having 20 stories.

g) **917942**- One definite red flag for this property was a very high value of AVTOT (actual total value of a property) when compared to the FULLVAL (total market value of a property), which seems very unusual from the pattern we found from the dataset. AVTOT is ~9 times higher than the FULLVAL. This opens opportunity for a possible investigation and try to find reason for this unusual behavior.

h) **85886**- Dimensions and Owner of the property imply that it's a park, that being said very high property costs compared to the land (which is situated near highway) should be investigated.

i) **585120-** This record corresponds to 20 stories building in NY with quite higher values of LTFRONT and LTDEPTH (139 ft and 342 ft respectively) and with very low property values. FULLVAL is around $2M which doesn't justify the characteristics of the property. So, this is a possible case of fraud by using framed AVTOT (actual total value of a property) and FULLVAL (total market value of a property). Also, values of BLDFRONT and BLDDPETH as 1 ft each don't match with this property having 20 stories.

j) **920628-** This record is a great example of where AVTOT (actual total value of a property) has been made up to ensure less tax payments. FULLVAL (total market value) is $ 1.9M, but AVTOT has a value of $ 0.075M which is highly unusual for a high value of FULLVAL. Also, BLDFRONT and BLDDEPTH both have values of 1 ft, but still AVTOT is 8 times higher than the AVLAND, which definitely raises a red flag and this record should be open for further investigation.

Part VII. Conclusions

Summary

In order to identify potential fraud records for more than 1 million New York City Properties, we followed an extensive fraud detection process driven by data processing and analytics including data cleaning, variable construction, dimensionality reduction, fraud algorithms' application and potential fraudulent records identification by business sense. Python, R, Advanced Excel and Tableau are used during the process as effective tools.

After NQR and NA filling, we looked for unusual valuations on the fields FULLVAL, AVLAND and AVTOT and built 45 special variables to as best as possible scale these value fields to look for anomalies; some variables use entities and groupings. After we had all the expert variables and their respective values at hand, we started the process of standardization and dimensionality reduction for further analysis. After Z-Scaling, we performed principal components analysis using the prcomp() function, looking at the eigenvalues and kept it within 70%.

We then chose Heuristic Function of z-scores and Auto encoders as our Fraud score algorithms to get two scores Score1 and Score2 respectively. After scoring all the records, we sort records respectively by 'Score 1&2' in ascending order and replace 'Score 1&2' with the sorted rank order (Lowest score getting rank 1 and so on). Finally, we took the average of Score 1 and Score 2 to get 'Cumulative score', which is our Final Fraud score.

High scored records are analyzed emphatically since they are highly suspicious fraudulent properties. We provided a detailed description analysis and also laid our special stress on analyzing the particularity of the top ten records, explaining their anomaly based on Full Value, Stories, Borough and also the owner. For those top property records with highest fraud score, we suggest a second assessment to prevent tax evasion. However, we still have many things that can be done in the future for this project.

Improvements

1. Filling 0 and NA are not enough

By analyzing the DQR, we find that for several important variables such as **LTFRONT** (Lot frontage(width) in feet), **LTDEPTH** (Lot depth in feet. LTDEPTH), **BLDDEPTH** (Building depth in feet) and **BLDFRONT** (Building front area in feet) there are also strange data entries except 0 and NA. Among the 10 records we found, more than half have **BLDDEPTH and BLDFRONT** as 1 and **LTFRONT and LTDEPTH** as 1, which is not statistically logical compared to their FULLVAL and STORIES. For future improvements, we may assume values for these four variables less than 5 should be treated as typos and also need to refill. Also, those 'unusual' entries should not be included in calculating mean/mode when grouping, because it can affect the whole data trend and summary of data. We also need to change those too-small values to mean or medium values.

2. More variables can be created

In our model, we created 9 key ratio variables that should be focused for risk valuation. We also identify three other ratios **possibly** significant for fraud detection by applying our risk score value:

a) FULLVAL / AVTOT: ratio of full market value of building to assessed value of property It's common to see FULLVAL twice as high as AVTOT in Metropolitan area like NYC, while a too large ratio compared to the mean will indicate the possibility of fraud.

b) AVTOT / EXTOT: the ratio of assessed value to exemption value of property

c) AVLAND / EXLAND: the ratio of assessed value to exemption value of land

3. Best Way to determine #PCA Components

In the PCA part, we can divide the variance explained by each PC by the total variance explained by all 45 PCs. compute the proportion of variance explained by each principal component, make the scree plot and cumulative plot to determine which PCs to keep; we would like to use the smallest number of PCs required to get a good understanding of the data. By examining the scree plot to see if there is a significant drop between two PC numbers, we can finally determine the number of PC components more accurately.

Appendix

Data Quality Report on NY property data

High-Level Description of Data

The Property Valuation and Assessment Data represents NYC properties assessments for purpose to calculate Property Tax, Grant eligible properties Exemptions and/or Abatements. Data collected and entered into the system by various City employee, like Property Assessors, Property Exemption specialists, ACRIS reporting, Department of Building reporting, etc.

The data was created on September 2, 2011 and updated annually. The final assessment time of this dataset is 2011-11-01.Therefore it covers various information of each property in New York City by November 1, 2011. There are 1070994 records and 32 fields in the dataset, while some information is nonindictable or missing(N/A).

According to the website, the latest version was released on September 10, 2018.

Summary Table of All fields

Numeric variable

Label	Numeric				% Populated	# Unique	# Zeros
	Min	Max	Avg	SD			
LTFRONT	0	9,999	36.63	74.03	100.0	1297	169108
LTDEPTH	0	9,999	88.9	76.4	100.0	1370	170128
STORIES	1	119	5.0	8.36	94.74	112	0
FULLVAL	0	6.15 e+09	8.74 e+05	1.16 e+07	100.0	109324	13007
AVLAND	0	2.66 e+09	8.50 e+04	4.05 e+06	100.0	70921	13009
AVLAND2	3	2.37 e+09	2.46 e+05	6.17 e+06	26.39	58592	0
AVTOT	0	4.67 e+09	2.27 e+05	6.87 e+06	100.0	112914	13007
AVTOT2	3	4.50 e+09	7.14 e+05	1.16 e+07	26.39	111361	0
EXLAND	0	2.67e+09	3.64 e+04	3.98 e+06	100.0	33419	491699
EXLAND2	1	2.37 e+09	3.51 e+05	1.08e+07	8.16	22196	0
EXTOT	0	4.67 e+09	9.12 e+04	6.51 e+06	100.0	64255	432572
EXTOT2	7	4.50 e+09	6.57 e+05	1.61e+07	12.21	48349	0
BLDFRONT	0	7.57 e+03	2.30 e+01	3.56 e+01	100.0	612	228815
BLDDEPTH	0	9.39 e+03	3.99 e+01	4.27 e+01	100.0	621	228853

Categorical variable

Label	Category		% Populated	# Unique	# Zeros
	Field Value	Most Common			
BBLE	AV-BORO’+ ‘AV_BLOCK’+ ‘AV_LOT ’	/	100.0	1079904	0
B	1 = MANHATTAN 2 = BRONX 3 = BROOKLYN 4 = QUEENS 5 = STATEN ISLAND	4	100.0	5	0
BLOCK	/	3944	100.0	13984	0
LOT	1 - 9978	1	100.0	6366	0
EASEMENT	/	E	43.28	13	0
OWNER	Personnel, Government, Business	PARKCHESTER PRESERVAT	97.03	863348	0
BLDGCL	/	R4	100.0	200	0
TAXCLASS	/	1	100.0	11	0
EXT	E/G/EG	G	33.08	3	0
EXCD1	1010-7170	1017	59.62	129	0
EXCD2	1011-7160	1017	8.67	61	0
STADDR	/	501 SURF AVENUE	99.93	839281	0
ZIP	10001-33803	10314	97.21	197	0
EXMPTCL	/	X1	1.45	14	0