DSO 562: FRAUD ANALYTICS

FINDING ANOMALIES IN NYC PROPERTY DATA

PREPARED BY TEAM 4:

CHUANG, HUNGLI
GUPTA, VARUN
HU, YITING
LIN, QIONGQIONG
RAWAT, AKASH
SRIVASTAVA, ASTHA
YU, SHUI

PREPARED FOR:

CITY OF NEW YORK
PROFESSOR STEPHEN COGGESHALL

USC University of Southern California

Table of Contents

Executive Summary

Objective

Our analytics team was hired by the city of New York to find anomalies in NY property data. We worked with New York City Property Data, consisting of over one million records with factors defining property valuation and assessment. This report explains in detail how we assessed the data using unsupervised machine learning methods in Python to detect the underlying anomalies. These anomalies can be investigated further to identify potential real estate fraud.

Project Outline

We analyzed the NY property data and then applied various methods to finalize upon the anomalies. The process in brief is explained below:

- Data cleaning: In this step, we filled the missing values with mean/median/mode of the group the missing value row belonged to. This ensured that the missing fields are filled with the most typical value
- Expert Variables: Then, we created special variables to as best as possible represent value fields to look for anomalies
- Standardization and dimensionality reduction: We scaled the values before applying Principal Component Analysis (PCA) to reduce dimensions. We scaled the PCA values again in this step
- Fraud Scores: We then combined these values to get first fraud score and reconstruction errors through autoencoder for second fraud score
- Ranking: We then calculated weighted rank average to get the final fraud score to be used for ranking potential anomalies in the data

Based on the final ranking, we identified top 10 anomalous records and could be investigated further for real estate fraud.

We observed and inferred from these records that these anomalies could exist as incorrect data was used as input, forged values of the properties were submitted or tax evasion.

Description of Data

Overall Description

Data represent NYC properties assessments for purpose to calculate 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, department of building reporting, et al. The following table describes the properties in detail -

Dataset Name:	Property Valuation and Assessment Data of Properties in NYC
Data Source:	Open Source of NYC government
Time Period:	November 2010
Number of Fields:	32
Number of Records:	1,070,994

Key Variables

We have described below the key variables. The detailed summary of the variables can be found in the appendix.

ZIP (Categorical, 5-digit code)

This field describes the zip code of the properties. We found that this column had 29,890 (2.79%) missing values and 196 unique values. We then looked at the top zip codes with maximum number of properties located within them –

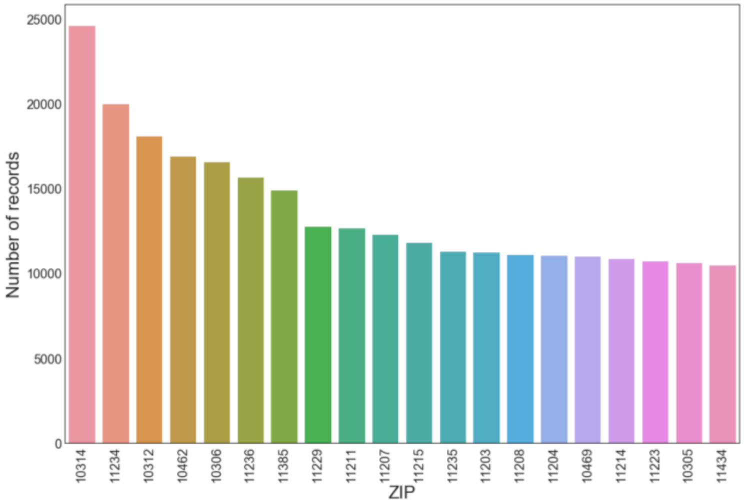


Figure 1.1 Distribution of ZIP

We can observe that zip code '10314' consists of the maximum number of properties.

B (Categorical, dtype: int64)

This categorical variable defines the Borough Codes with the following definitions

- 1 = MANHATTAN
- 2 = BRONX
- 3 = BROOKLYN
- 4 = QUEENS
- 5 = STATEN ISLAND

The following graph shows the spread of NY properties across these Boroughs.

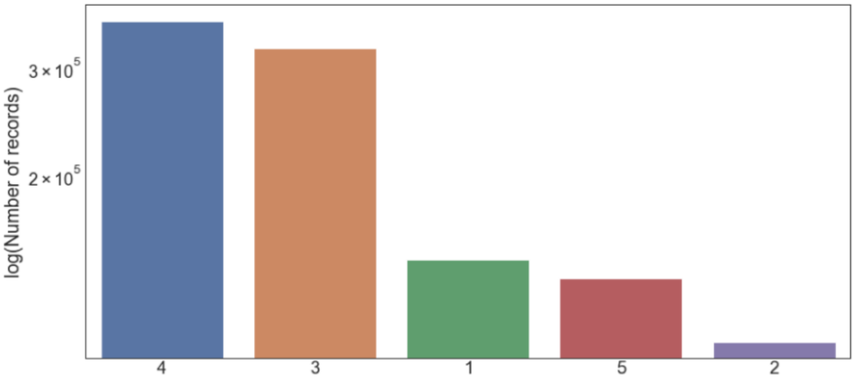


Fig 1.2 Categorical distribution of 'B' variable

We can observe that maximum number of properties are located in Queens and Brooklyn in the data.

FULLVAL (Numeric, dtype: float64)

This field describes the total market value of the property in dollars. We found that this column did not have any missing values, however contained 13,007 properties (1.21%) with 0 value. The field statistics of the variable (excluding 0 values) are as following –

Count	1057987
Mean	885012.8
Standard Deviation	11653000
Minimum	4
25%	311000
50%	450000
75%	623000
Maximum	6150000000

Table 1.1 Statistics of the variable

We can still observe that the minimum value is 4 which is highly unlikely. The values could be further cleaned if required. Also, 75% of the properties have value less than 650,000. Hence, we looked at the distribution of the variable by capping the maximum value at 1,500,000 –

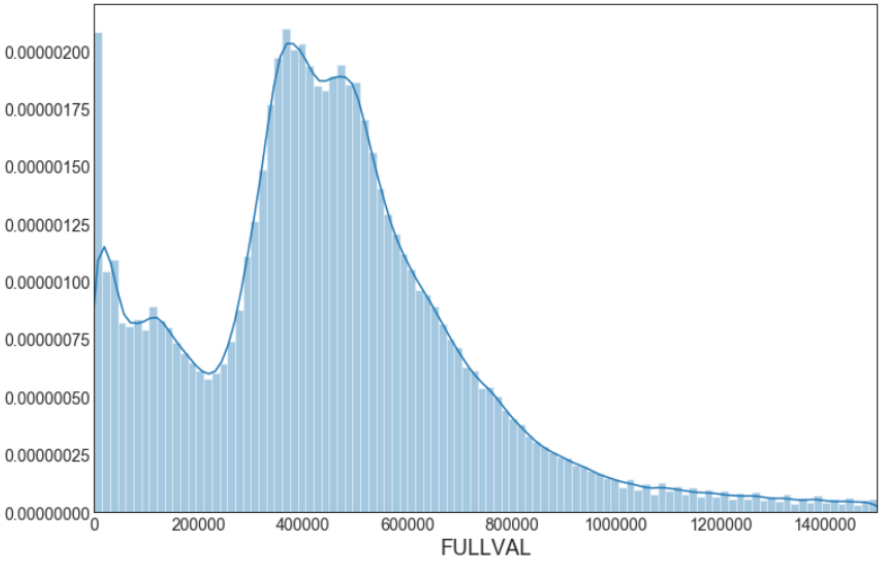


Fig 1.3 Density plot of the variable

We can observe that the number of properties declines exponentially after a threshold as the FULLVAL value increases.

AVLAND (Numeric, dtype: float64)

This field describes the actual land value of the property in dollars. We found that this column did not have any missing values, however contained 13,009 properties (1.21%) with 0 value. The field statistics of the variable (excluding 0 values) are as following –

Count	1057985
Mean	86113.92
Standard Deviation	4082117
Minimum	1
25%	9445
50%	13782
75%	19860
Maximum	2668500000

Table 1.2 Statistics of the variable

We can still observe that the minimum value is 1 which is highly unlikely. The values could be further cleaned if required. Also, 75% of the properties have value less than 20,000. Hence, we looked at the distribution of the variable by capping the maximum value at 50,000 –

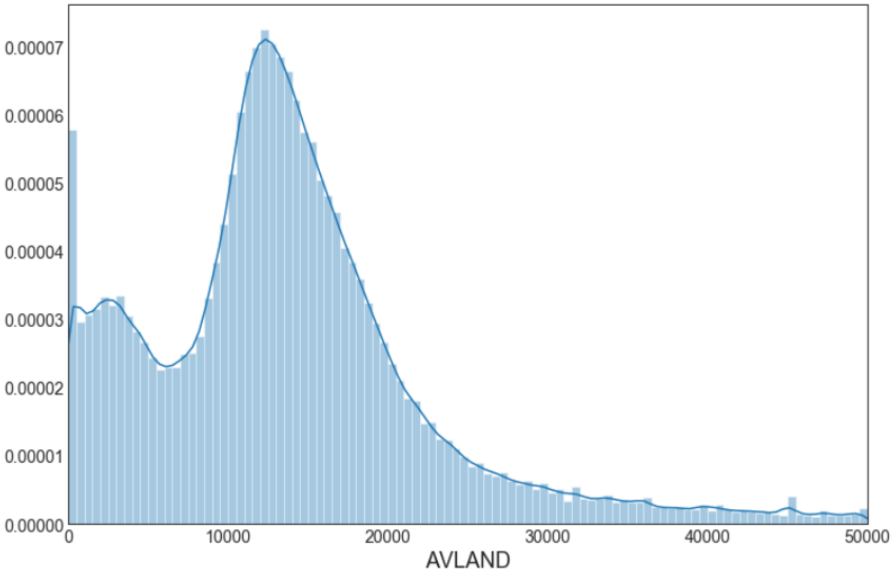


Fig 1.4 Density plot of the variable

We can observe that the number of properties declines exponentially after a threshold as the AVLAND value increases.

AVTOT (Numeric, dtype: float64)

This field describes the actual land value of the property in dollars. We found that this column did not have any missing values, however contained 13,007 properties (1.21%) with 0 value. The field statistics of the variable (excluding 0 values) are as following –

Count	1057987
Mean	230031.9
Standard Deviation	6919630
Minimum	1
25%	18657
50%	25560
75%	46250
Maximum	4668309000

Table 1.3 Statistics of the variable

We can still observe that the minimum value is 1 which is highly unlikely. The values could be further cleaned if required. Also, 75% of the properties have value less than 50,000. Hence, we looked at the distribution of the variable by capping the maximum value at 100,000 –

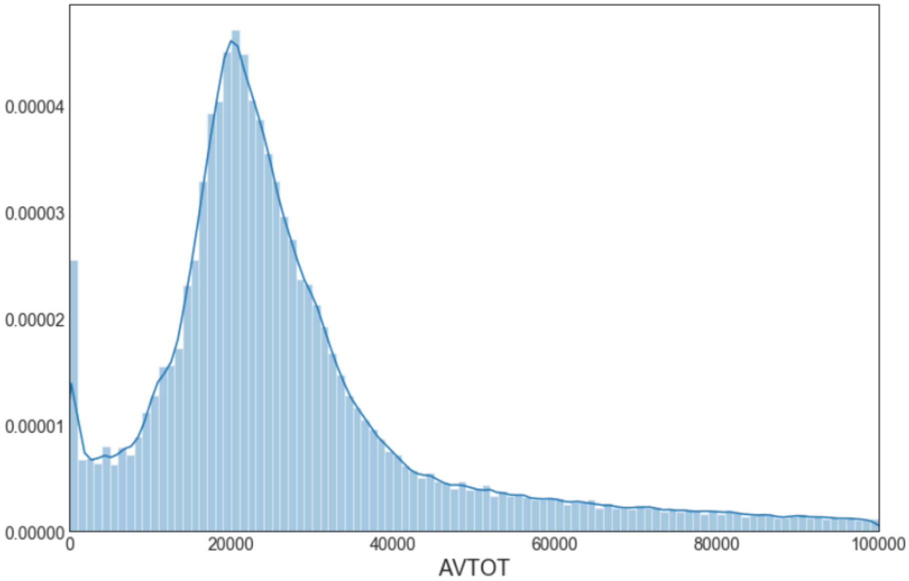


Fig 1.5 Density plot of the variable

We can observe that the number of properties declines exponentially after a threshold as the AVTOT value increases.

LTFRONT (Numeric, dtype: int64)

This field describes the lot front of the property in feet. We found that this column did not have any missing values, however contained 169108 (15.79%) properties with 0 values. The field statistics of the variable are as following –

Count	1070994
Mean	36.6353
Standard Deviation	74.03284
Minimum	0
25%	19
50%	25
75%	40
Maximum	9999

Table 1.4 Statistics of the variable

75% of the properties have value less than 40. Hence, we looked at the distribution of the variable by capping the maximum value at 100 –

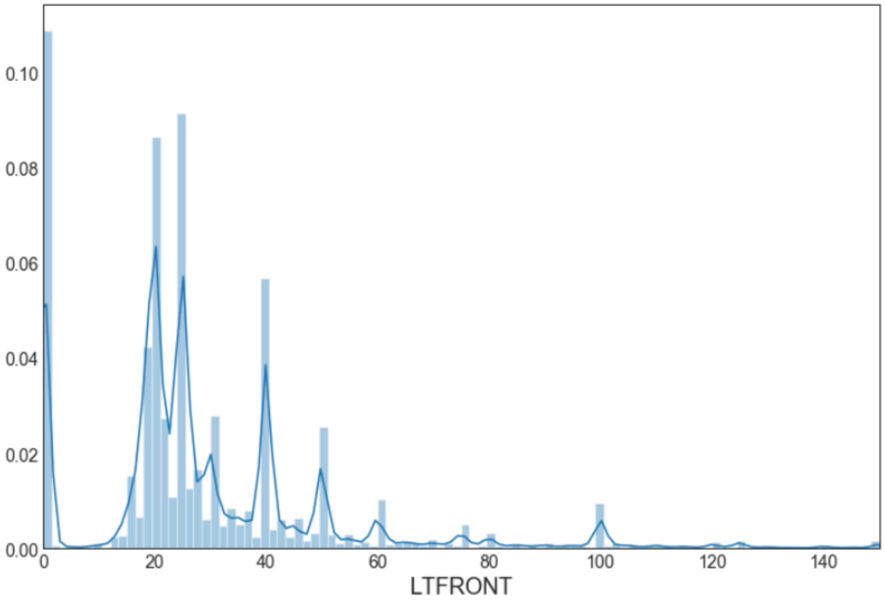


Fig 1.6 Density plot of the variable

LTDEPTH (Numeric, dtype: int64)

This field describes the lot depth of the property in feet. We found that this column did not have any missing values, however contained 170128 (15.89%) properties with 0 values. The field statistics of the variable are as following –

Count	1070994
Mean	88.86159
Standard Deviation	76.39628
Minimum	0
25%	80
50%	100
75%	100
Maximum	9999

Table 1.5 Statistics of the variable

More than 75% of the properties have value less than 150. Hence, we looked at the distribution of the variable by capping the maximum value at 200 –

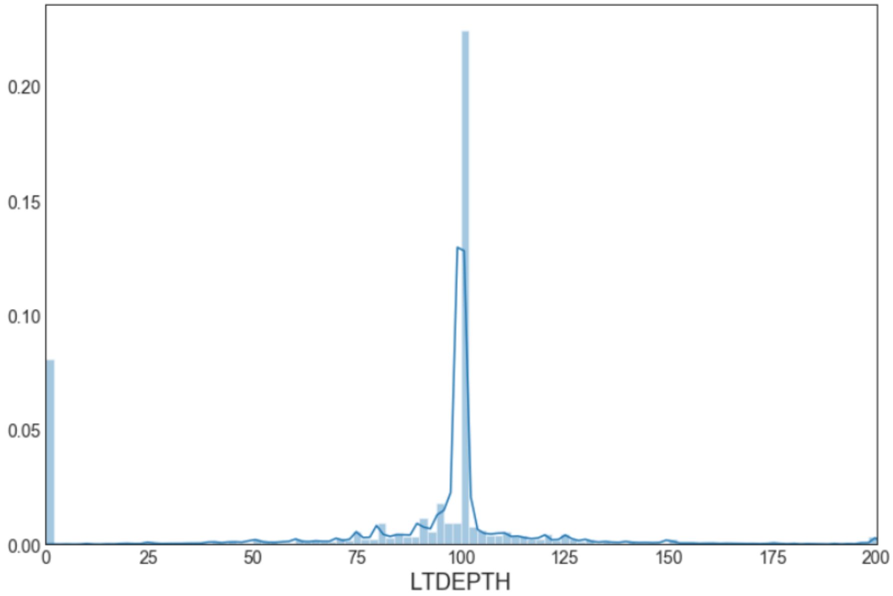


Fig 1.7 Density plot of the variable

We can observe that maximum number of properties have LTDEPTH value as 100.

BLDFRONT (Numeric, dtype: int64)

This field describes the building front of the property in feet. We found that this column did not have any missing values, however contained 228815 (21.36%) properties with 0 values. The field statistics of the variable are as following –

Count	1070994
Mean	23.04277
Standard Deviation	35.5797
Minimum	0
25%	15
50%	20
75%	24
Maximum	7575

Table 1.6 Statistics of the variable

More than 75% of the properties have value less than 30. Hence, we looked at the distribution of the variable by capping the maximum value at 50 –

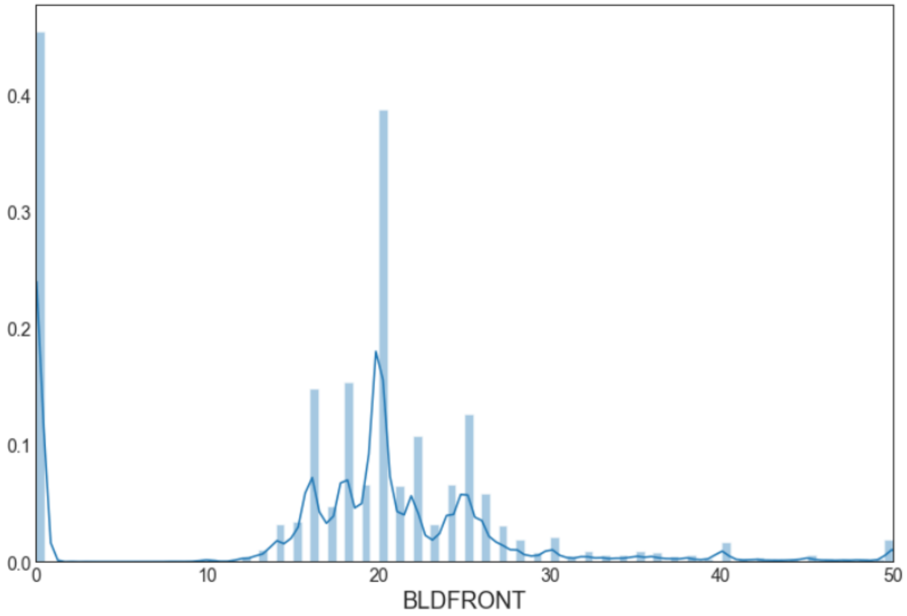


Fig 1.8 Density plot of the variable

We can observe that maximum number of properties have BLDFRONT value in the range of 10-30.

BLDDEPTH (Numeric, dtype: int64)

This field describes the building depth of the property in feet. We found that this column did not have any missing values, however contained 228853 (21.37%) properties with 0 values. The field statistics of the variable are as following –

Count	1070994
Mean	39.92284
Standard Deviation	42.70715
Minimum	0
25%	26
50%	39
75%	50
Maximum	9393

Table 1.7 Statistics of the variable

More than 75% of the properties have value less than 70. Hence, we looked at the distribution of the variable by capping the maximum value at 120 –

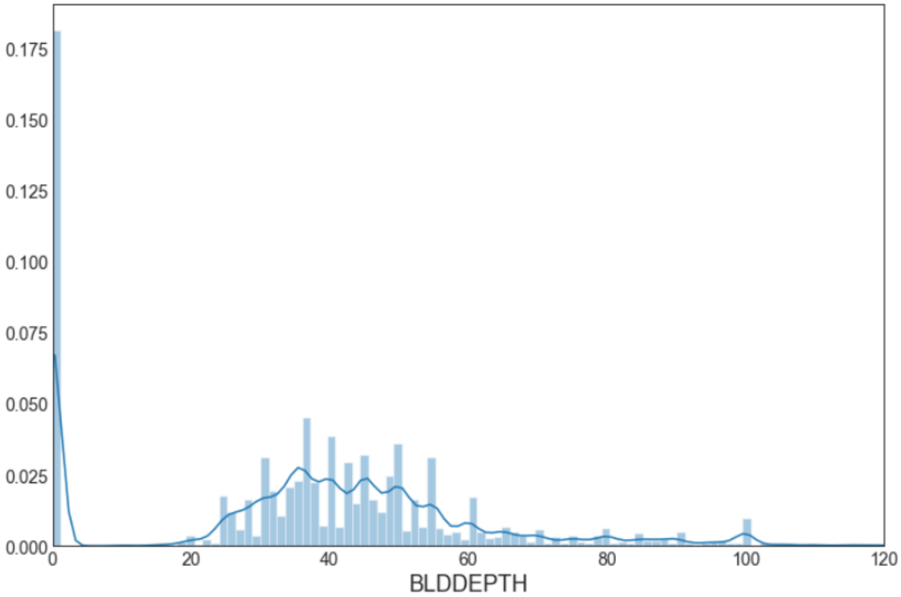


Fig 1.9 Density plot of the variable

We can observe that maximum number of properties have BLDFRONT value in the range of 10-30.

TAXCLASS (Categorical, 2-letter string)

This field describes the tax class of the properties. We found that this column had 11 unique values and no missing values. We then looked at the distribution of the properties among these 11 tax classes –

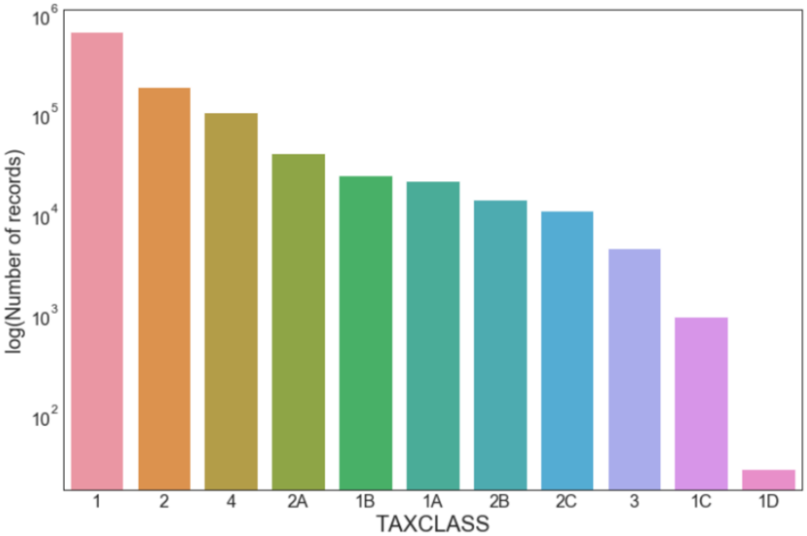


Fig 1.10 Distribution of the TAXCLASS variable

We can observe that maximum number of properties lie within tax class 1.

STORIES (Numeric, dtype: int64)

This field describes the number of stories in the property. We found that this column contained 56264 (5.25%) properties with null/missing values. The field statistics of the variable are as following –

Count	1014730
Mean	5.006918
Standard Deviation	8.365707
Minimum	1
25%	2
50%	2
75%	3
Maximum	119

Table 1.8 Statistics of the variable

More than 75% of the properties have value less than 5. Hence, we looked at the distribution of the variable by capping the maximum value at 50 (to consider outliers –

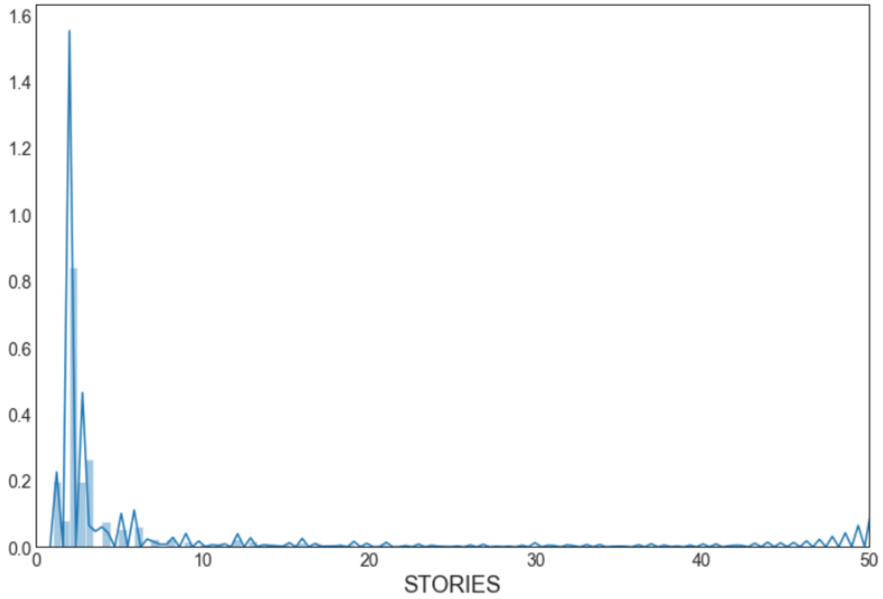


Fig 1.11 Density plot of the variable

We can observe that maximum number of properties have number of STORIES in the range of 1-10. However, we can also observe potential outliers with number of stories around 50.

Data Cleaning

Filling ZIP

1. For NULLs or 0's, group by B, 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 groupings in which all the ZIP are missing, run a function to return the nearest non-NA value to the rest of NAs (e. g. For BLOCK 205, if all ZIP in BLOCK 204 and 206 are 90007, fill the BLOCK 205 with 90007, too).

Compare the frequency of group one below and one above, insert the ZIP value of the group with higher frequency.

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 B, 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 variable, then filling the remaining variables with the median value of the groups.

Filling STORIES

1. For NULLs or 0’s, group by ZIP and BLDGCL, calculate and fill with median value of group.

2. For remaining NULLs or 0’s, group by BLDGCL and STORIES, calculate and fill with median value of group.

3. For remaining NULLs or 0’s, group by TAXCLASS and STORIES, calculate and fill with median value of group.

4. For remaining NULLs or 0’s, group by STORIES, 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 ZIP and BLDGCL to create our first layer, followed by combinations of BLDGCL and STORIES, STORIES and TAXCLASS and finally simply STORIES as our backstop.

Variable Creation

After cleaning data and filling in missing fields, we created 3 size and 9 ratio variables in order to create more reasonable variables which can be used to detect anomalies. Then, we created 45 new variables by grouping each ratio variable into 5 different groups – ZIP, ZIP3, TAXCLASS, BORO and ALL.

Step 1 Create 3 size variables

We chose three important variables from the dataset to base our new variables on –

1) V_1, FULLVAL: Full market value of the property
2) V_2, AVLAND: Actual land value of the property (doesn't include the construction)
3) V_3, AVTOT: Actual total value of a property/land/ other structures (including the construction cost)

After this we created three new size variables which could be further used to create new variables. Following are the formulas for the three size variables –

1) S_1, Lot Area = Lot Frontage * Lot Depth (LTFRONT * LTDEPTH)
2) S_2, Building Area = Building Frontage * Building Depth (BLDFRONT * BLDDEPTH)
3) S_3, Building Volume = Building Area * Stories (S_2 * STORIES)

Step 2 Create key ratio variables

In the next step, we created 9 key ratio variables by using three important variables (V_1, V_2 and V_3) and three new size variables (S_1, S_2 and S_3). V_1, V_2 and V_3 were used as denominators, while V_4, V_5 and V_6 were used as the denominators to create the 9 key ratio variables. We have listed all the combinations below –

$$\begin{array}{lll} r_1 = \frac{V_1}{S_1} & r_4 = \frac{V_2}{S_1} & r_7 = \frac{V_3}{S_1} \\ r_2 = \frac{V_1}{S_2} & r_5 = \frac{V_2}{S_2} & r_8 = \frac{V_3}{S_2} \\ r_3 = \frac{V_1}{S_3} & r_6 = \frac{V_2}{S_3} & r_9 = \frac{V_3}{S_3} \end{array}$$

Step 3 Create 45 expert variables

Finally, to ensure that the records with anomalies (outliers) are more noticeable, we chose five variables to group the values with. The five groups we selected were –

- ZIP
- ZIP3
    - ZIP3 represents the zip code with the same first three digits. We created this variable by extracting the first three digits from the original 5-digit zip code

- TAXCLASS
- BORO ('B')
- ALL
    - The group "All" means there is no specific group and we use the entire dataset as one group.

We used these five variables to ensure that properties within same vicinity and tax class are not treated as anomalies when compared with properties from another location or tax class. For instance, properties in Brooklyn and New York would be priced differently.

We then calculated mean of all the nine ratio variables created above for each group based on these five categories. Then we divided the nine ratios of each row/property with the mean of each group. We can represent the 45 expert variables through –

$$\frac{r_1}{< r_1 >_g}, \quad \frac{r_2}{< r_2 >_g}, \quad \frac{r_3}{< r_3 >_g}, \quad \cdots \quad \frac{r_9}{< r_9 >_g}$$

Where g represents each of the five groupings mentioned before.
To summarize the variable creation process, we used three important variables and three size variables to create nine ratios. These ratios were then grouped using five categories and their mean was calculated. Lastly, we divided each ratio with the mean of each group to create 45 expert variables.

The list of all the variables can be found in the appendix.

Dimensionality Reduction

Overview

After creating the expert variables, our next step was to reduce the dimensionality by performing Principal Component Analysis (PCA). However, the original expert variables had different scales, we couldn't directly perform PCA based on these variables. We z-scaled the expert variables using sklearn library in python.

PCA is a statistical technique to extract patterns and remove dimensionality in the dataset. While we created the expert variables, there would be too many features or variables to process. Therefore, after we normalized the data, PCA was applied to extract new independent principal components (PC's). PCA calculates the eigenvectors of covariance matrix of data to find the patterns and orders all PC's in decreasing order of their eigenvalues. We kept the PC's with higher eigenvalues to be our new features and dropped the rest of them to reduce the dimensionality. This new data has less features than the old one but still contains most of the information.

Lastly, we wanted all the principal components to be equally scaled, so we z-scaled again to normalize the PC's before we conducted the following algorithms.

Results

As a result, we fitted the PCA algorithm into our data, and plot the cumulative summation of the explained variance using scree plot.

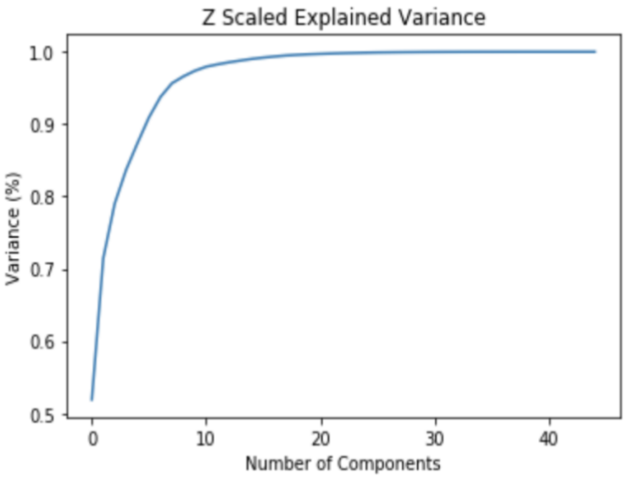
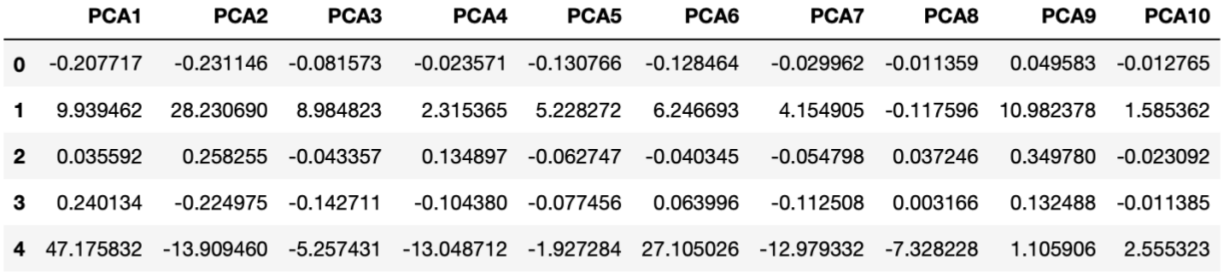


Fig 1.12 Scree plot of the PCA

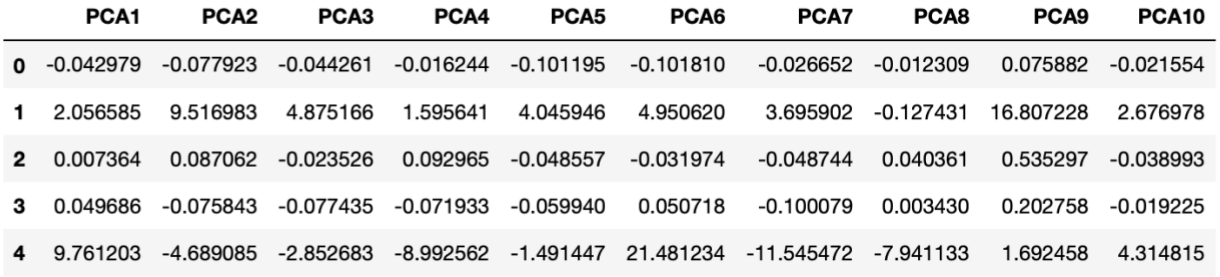
Based on the explained variances, we decided to keep the top 10 principal components to perform our further analysis as it explained more than 95% of the variance.



	PCA1	PCA2	PCA3	PCA4	PCA5	PCA6	PCA7	PCA8	PCA9	PCA10
0	-0.207717	-0.231146	-0.081573	-0.023571	-0.130766	-0.128464	-0.029962	-0.011359	0.049583	-0.012765
1	9.939462	28.230690	8.984823	2.315365	5.228272	6.246693	4.154905	-0.117596	10.982378	1.585362
2	0.035592	0.258255	-0.043357	0.134897	-0.062747	-0.040345	-0.054798	0.037246	0.349780	-0.023092
3	0.240134	-0.224975	-0.142711	-0.104380	-0.077456	0.063996	-0.112508	0.003166	0.132488	-0.011385
4	47.175832	-13.909460	-5.257431	-13.048712	-1.927284	27.105026	-12.979332	-7.328228	1.105906	2.555323

Table 1.9 PCA1-10 values

Lastly, we applied z-scale again on these PCA values to normalize them.



	PCA1	PCA2	PCA3	PCA4	PCA5	PCA6	PCA7	PCA8	PCA9	PCA10
0	-0.042979	-0.077923	-0.044261	-0.016244	-0.101195	-0.101810	-0.026652	-0.012309	0.075882	-0.021554
1	2.056585	9.516983	4.875166	1.595641	4.045946	4.950620	3.695902	-0.127431	16.807228	2.676978
2	0.007364	0.087062	-0.023526	0.092965	-0.048557	-0.031974	-0.048744	0.040361	0.535297	-0.038993
3	0.049686	-0.075843	-0.077435	-0.071933	-0.059940	0.050718	-0.100079	0.003430	0.202758	-0.019225
4	9.761203	-4.689085	-2.852683	-8.992562	-1.491447	21.481234	-11.545472	-7.941133	1.692458	4.314815

Table 1.10 Z scaled PCA values

Algorithms

After conducting PCA, we reduced the dimensionality of our data from $\mathrm{m} = 45$ to $\mathrm{p} = 10$ and standardize the data using z-scaling. Then, we used two methods to calculate two fraud scores for each record, respectively heuristic function and autoencoder. Finally, we combined the two scores to get the final fraud score and determine the fraud records.

Fraud Score 1: Heuristic Function

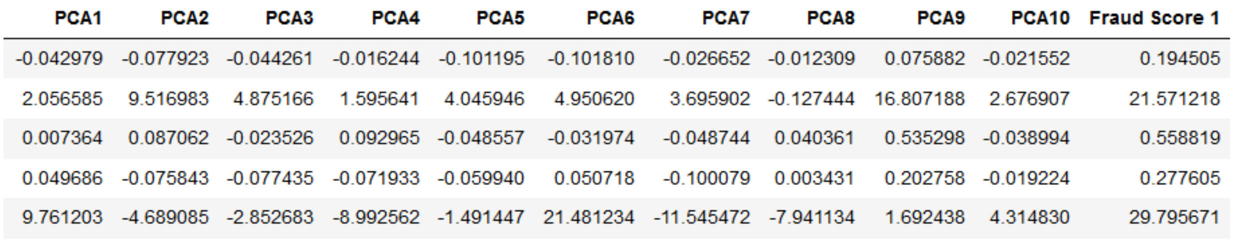
First, we used the Minkowski distance as the heuristic function to calculate the first fraud score $\mathrm{S_i}$. The heuristic function is

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

where k takes value from 1 to 10. Here, we used Minkowski distance with $\mathrm{n} = 2$, which corresponds to the Euclidean distance.

The first fraud score $\mathrm{S_i}$ describes how far each record is away from the mean of the whole dataset. Therefore, if a record has a high fraud score, this record is far away from most of the data, which means it is an outlier and a potential fraud record.

Snippet of the Fraud Score 1 –



PCA1	PCA2	PCA3	PCA4	PCA5	PCA6	PCA7	PCA8	PCA9	PCA10	Fraud Score 1
-0.042979	-0.077923	-0.044261	-0.016244	-0.101195	-0.101810	-0.026652	-0.012309	0.075882	-0.021552	0.194505
2.056585	9.516983	4.875166	1.595641	4.045946	4.950620	3.695902	-0.127444	16.807188	2.676907	21.571218
0.007364	0.087062	-0.023526	0.092965	-0.048557	-0.031974	-0.048744	0.040361	0.535298	-0.038994	0.558819
0.049686	-0.075843	-0.077435	-0.071933	-0.059940	0.050718	-0.100079	0.003431	0.202758	-0.019224	0.277605
9.761203	-4.689085	-2.852683	-8.992562	-1.491447	21.481234	-11.545472	-7.941134	1.692438	4.314830	29.795671

Table 1.11 Calculated Fraud Score 1 values

Distribution of Fraud Score 1 –

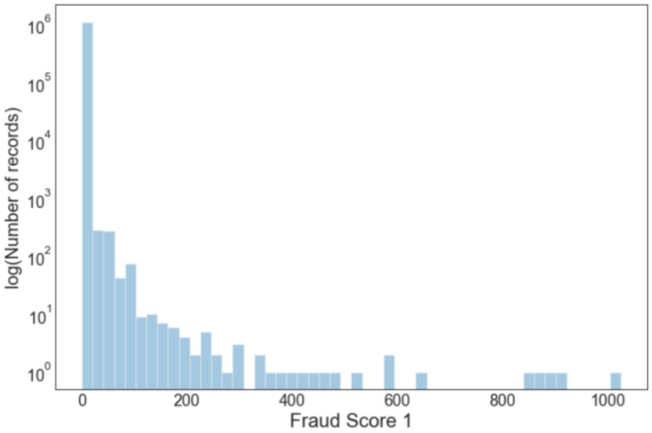


Fig 1.13 Histogram of Fraud Score 1

Fraud Score 2: Autoencoder

An autoencoder neural network is an unsupervised learning algorithm that sets the target values to be equal to the inputs. In other words, it is trying to learn an approximation to the identity function, so as to output $\hat{x}$ is similar to the input x.

Autoencoders are good at data denoising, which enables it to discover the unusual structures of the dataset. If the output is far away from the original input, it means that this record is a potential ‘noise’, which is the fraud record in our case. So, we used the distance between the outputs and inputs as our second fraud score $\widetilde{S_\imath}$, which is calculated as

$$\widetilde{S_\imath} = (\sum_{j=1}^{p} |\widehat{z_{\imath\jmath}} - z_{ij}|^{n})^{1/n} \quad (i \; = \; 1, \ldots, N)$$

where z_{ij} $(j = 1, \ldots, p)$ is the element of each z-scaled record $Zi = (z_{i1}, \ldots, z_{ip})$ $(i \; = \; 1, \ldots, N)$ and $\widehat{z_{\imath\jmath}}$ is the output of the autoencoder for z_{ij} . Similarly, we used Minkowski distance with $n = 2$, which corresponds to the Euclidean distance. With the second fraud score defined, the next step is to build the autoencoder.

Using Keras in Python, we build an autoencoder with the following structure –

- First, we imported the required ‘tensorflow’ library, from which we imported ‘keras’ to structure our autoencoder
- We used Rectified Linear Units (ReLu) and hyperbolic tangent (tanh) as activation functions for our hidden layers. These functions were first used to encode and then decode the input layer
- Then, we used tanh function again to output the final output values
- We trained our autoencoder by splitting the z scaled PCA values in the ratio of 8:2 to training and test data set

- We used 'tensorboard' and 'checkerpoint' to save the best autoencoder model while training over the dataset
- Lastly, we predicted the autoencoder values by running the best model on all of the input data (z scaled PCA values)

Below is a broad level representation of the autoencoder –

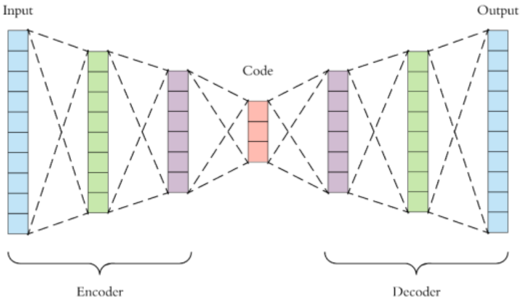
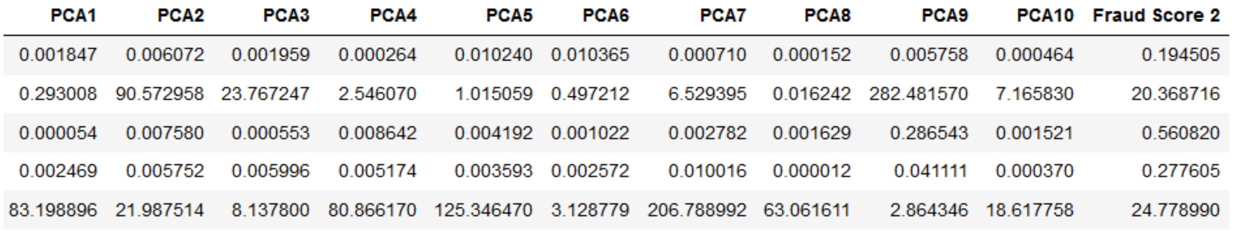


Fig 1.14 Autoencoder Representation

After obtaining the output, we calculated the second fraud score for each record using the formula mentioned before.

A snippet of the Fraud Score 2 –



PCA1	PCA2	PCA3	PCA4	PCA5	PCA6	PCA7	PCA8	PCA9	PCA10	Fraud Score 2
0.001847	0.006072	0.001959	0.000264	0.010240	0.010365	0.000710	0.000152	0.005758	0.000464	0.194505
0.293008	90.572958	23.767247	2.546070	1.015059	0.497212	6.529395	0.016242	282.481570	7.165830	20.368716
0.000054	0.007580	0.000553	0.008642	0.004192	0.001022	0.002782	0.001629	0.286543	0.001521	0.560820
0.002469	0.005752	0.005996	0.005174	0.003593	0.002572	0.010016	0.000012	0.041111	0.000370	0.277605
83.198896	21.987514	8.137800	80.866170	125.346470	3.128779	206.788992	63.061611	2.864346	18.617758	24.778990

Table 1.12 Calculated Fraud Score 2 values using autoencoder

Distribution of the autoencoder scores –

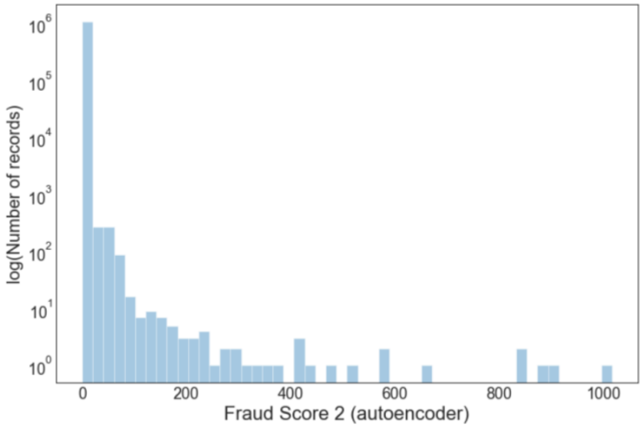


Fig 1.15 Histogram of Fraud Score 2

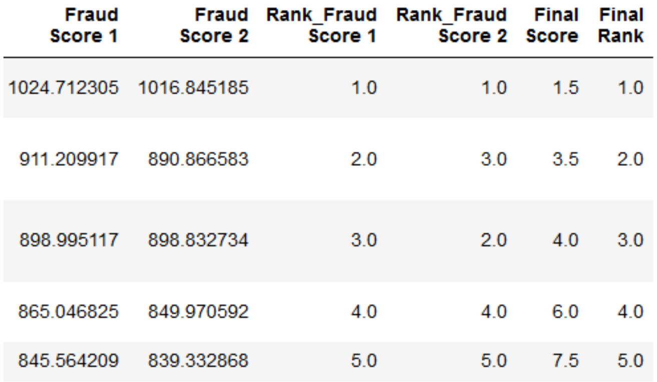
Final Fraud Score

With the two fraud scores calculated, we would combine them to produce the final fraud score. Noticing that there are differences between the two fraud scores, sometimes the difference could be very tiny while sometimes the difference is rather large. To incorporate this information, we would use the rank of each record by two fraud scores.

Our methodology to create the final score can be summarized as follows:

1. Rank the records by the first fraud score S_i and obtain rank of each record R_i $(i = 1, \dots, N)$
2. Rank the records by the second fraud score $\widetilde{S_i}$ and obtain rank of each record $\widetilde{R_i}$ $(i = 1, \dots, N)$
3. Calculate the final score F_i of record i as $F_i = (S_i \times R_i + \widetilde{S_i} \times \widetilde{R_i})/2$

Finally, we ranked the records using the final fraud score and were able to identify the frauds in our dataset, which were those who have the highest scores. A snippet of the final score and ranks –



Fraud Score 1	Fraud Score 2	Rank_Fraud Score 1	Rank_Fraud Score 2	Final Score	Final Rank
1024.712305	1016.845185	1.0	1.0	1.5	1.0
911.209917	890.866583	2.0	3.0	3.5	2.0
898.995117	898.832734	3.0	2.0	4.0	3.0
865.046825	849.970592	4.0	4.0	6.0	4.0
845.564209	839.332868	5.0	5.0	7.5	5.0

Table 1.13 Snippet of the final fraud score and ranks

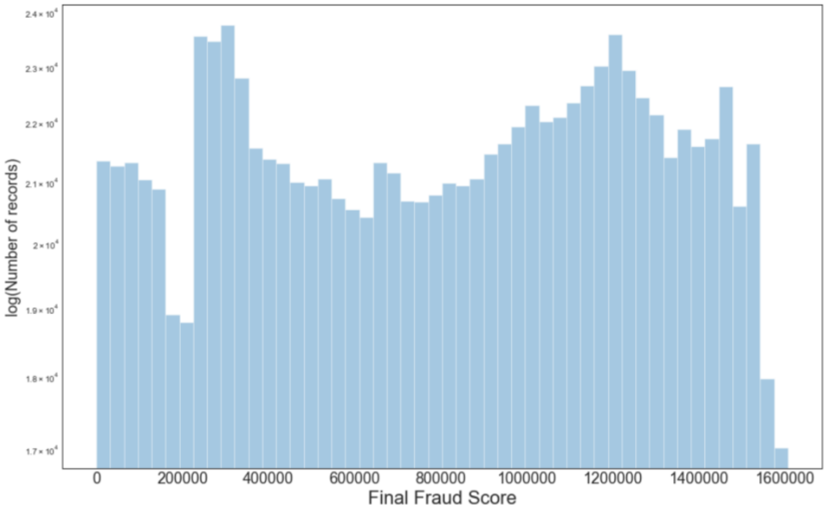
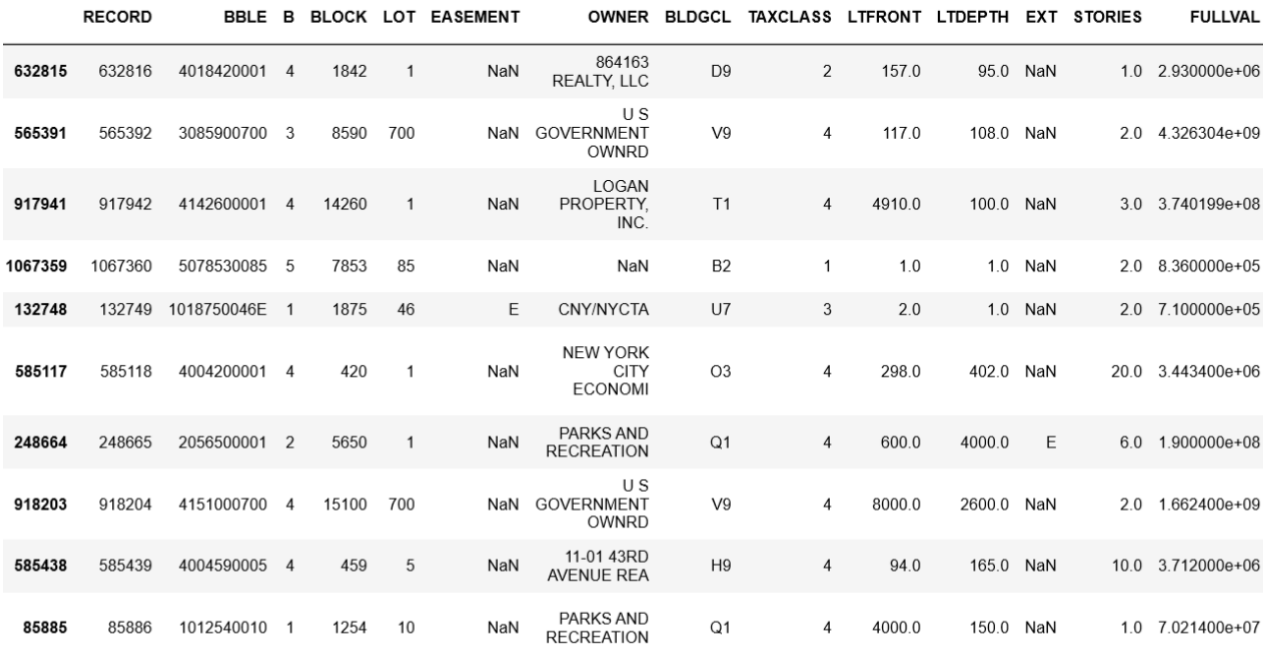


Fig 1.16 Histogram of the Final Fraud Score

Results

The Top ten records with the highest Cumulative average of Heuristic Fraud scores and Autoencoder Fraud scores are as follows:



	RECORD	BBLE	B	BLOCK	LOT	EASEMENT	OWNER	BLDGCL	TAXCLASS	LTFRONT	LTDEPTH	EXT	STORIES	FULLVAL
632815	632816	4018420001	4	1842	1	NaN	864163 REALTY, LLC	D9	2	157.0	95.0	NaN	1.0	2.930000e+06
565391	565392	3085900700	3	8590	700	NaN	U S GOVERNMENT OWNRD	V9	4	117.0	108.0	NaN	2.0	4.326304e+09
917941	917942	4142600001	4	14260	1	NaN	LOGAN PROPERTY, INC.	T1	4	4910.0	100.0	NaN	3.0	3.740199e+08
1067359	1067360	5078530085	5	7853	85	NaN	NaN	B2	1	1.0	1.0	NaN	2.0	8.360000e+05
132748	132749	1018750046E	1	1875	46	E	CNY/NYCTA	U7	3	2.0	1.0	NaN	2.0	7.100000e+05
585117	585118	4004200001	4	420	1	NaN	NEW YORK CITY ECONOMI	O3	4	298.0	402.0	NaN	20.0	3.443400e+06
248664	248665	2056500001	2	5650	1	NaN	PARKS AND RECREATION	Q1	4	600.0	4000.0	E	6.0	1.900000e+08
918203	918204	4151000700	4	15100	700	NaN	U S GOVERNMENT OWNRD	V9	4	8000.0	2600.0	NaN	2.0	1.662400e+09
585438	585439	4004590005	4	459	5	NaN	11-01 43RD AVENUE REA	H9	4	94.0	165.0	NaN	10.0	3.712000e+06
85885	85886	1012540010	1	1254	10	NaN	PARKS AND RECREATION	Q1	4	4000.0	150.0	NaN	1.0	7.021400e+07

Table 1.14 Top 10 records with highest fraud ranks

The methodology we followed to reach to the above records was based on the values of some important independent variables (B, BLOCK, LOT, TAXCLASS, LRFRONT, LTDEPTH, STORIES, FULLVAL, AVLAND, AVTOT, BLDFRONT, BLDDEPTH) from our dataset and use these variables to identify the fraudulent activities.

A quick recap to the most important variables in our book:

- FULLVAL - Full market value of the property
- AVLAND - Actual land value of the property (doesn't include the construction)
- AVTOT – Actual total value of a property/land/ other structures (including the construction cost)

Based on the above variables, the top ten records were reported. Here is a detailed analysis of the top ten records:

632816

This might be a case of forging property attributes to avoid the high payment of taxes. As we can see that based on various property details like TAXCLASS (since it is 2, hence, it is an apartment) and owner name, it is a commercial building with apartments in it, but extremely low values of BLDFRONT and BLDDEPTH look suspicious. It might be possible that the building is still under construction but property like these is always open to investigation.

565392

Given the available land area, property has a very high valuation ($4 B) which is evident from the values of LTFRONT (~117) and LDEPTH (~108). The pattern of such record is an outlier, and should be categorized 'susceptible', and investigated for fraud.

917942

The property has a high value of AVTOT (total actual value of a property) as compared to FULLVAL (total market value of a property), which seems very different from other observations of the dataset. Statistically, AVTOT is ~9 times higher than the FULLVAL which opens opportunity for a possible investigation of this unusual behavior.

1067360

There are two reasons two reasons which might contribute to the flagging of this record.

1) High difference between AVTOT and FULLVAL. The FULLVAL is around 16 times higher than AVTOT.

2) The property's dimensions don't make logical sense when comparing high values of BLDFRONT and BLDDEPTH as compared to LTFRONT and LTDEPTH.

132749

The property's dimensions don't make logical sense when comparing high values of BLDFRONT and BLDDEPTH as compared to LTFRONT and LTDEPTH. Also, the value of BLDDEPTH is 100 which is much greater for a 2-story building.

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 FULLVAL at $3.5M which is
quite low as compared to the average property value of a 20-story building in NYC. Also,
values of BLDFRONT and BLDDPETH are 1 foot each, which doesn't make logical sense for a twenty-floor building.

248665

The LTDEPTH of this building is very high (~400) as compared to the AVTOT and FULLVAL value of the same. Hence, the property evaluation has been maintained low to invoke tax fraud in the future. This building should be investigated.

918204

This is a government owned property with very high values of both LTFRONT(~8000) and LTDEPTH (~2600) which means that the coverage is more than the average properties of NYC but the same result is not reflected in the BLDFRONT and BLDDEPTH which are 20 and 38. Also, the area pin code is 1 which is invalid in the USA. Hence, an investigation is needed in this property suspected of fraud.

585439

The building is a 10-story building with FULLVAL as low as $ 3.7 M, hence it creates a red flag. Also, AVTOT is 16 times as AVLAND but there is not much construction on the land (since BLDFRONT and BLDDEPTH are 1 foot each). Hence, this might also be a classic example of an attempt to tax fraud.

85886

The Owner of the property states that it is a park and recreation center which is supported by the large area of the property. However, the number of stories in this property is just 1 which is not consistent with the park and recreation center category. Also, this property has been flagged due to high LTFRONT and LTDEPTH values which are inconsistent with the properties in its vicinity.

Conclusions

In order to detect potential fraud records for observations of more than 1 million New York City Properties, we followed a detailed fraud identification process driven by data processing and analytics including the data cleaning, creation of expert variables, dimensionality reduction, scoring by two methods, combining the scores using weighted average rank orders, and finally applying business sense to understand the fraud trends.

We have used Python for completing the project and concluding results. The first step was to impute the missing value of key variables like ZIP, FULLVAL, STORY etc. We further built 45 expert variables using various combinations and grouping techniques. 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.

We then chose Heuristic Function of z-scores using Euclidean distance, and autoencoder neural network as our Fraud score algorithms to get the two scores 'Fraud Score 1' and 'Fraud Score 2' respectively. After scoring all the records, we rank each fraud score, and evaluate the final score by averaging the ranks of each property.

Finally, the records with high fraud scores have been analyzed, and suspected as suspicious properties. In the end, we have provided a detailed reasoning as to why the top-10 high score properties may be fraudulent cases based on our business sense.

Potential Improvements

We believe following areas could be looked upon for improvement –

- Outlier values
  We have in the current analyses filled 0 or NA values. However, we have not dealt with the present outliers in the dataset. For instance, there are properties which have low LTDEPTH, LTFRONT, BLDFRONT and BLDDEPTH (~1) which is inconsistent with their FULLVAL evaluation or the number of stories.
- Create more expert variables
  Currently, we have not used the EXTOT and EXLAND values which refer to the exemption value of the property and exemption value of the land. We could combine these variables with FULLVAL and AVLAND to create ratios and identify any outlier values