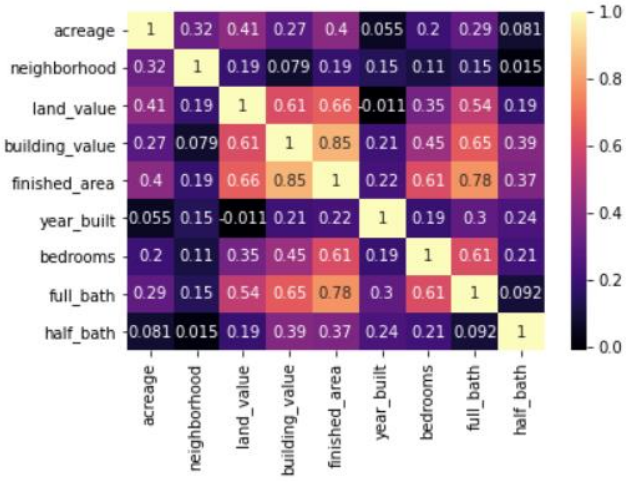
Introduction

The dataset contains property sales data in Nashville area. The objective of this project is to build a machine learning model for a real estate company which can be used to accurately find the best value deals in the Nashville area. I'll use various python packages such as numpy, pandas, sklearn, statsmodel to analyze and find patterns the data set. The business problem of the project is to find out the key factors that lead to a best deal for the company. Using variable 'Sale Price Compared to Value' we'll determine whether a property is undervalued or overvalued. With the help of predictive model, the real estate company can target those properties which are undervalued and deliver the maximum value in the deal. I'll create predictive models using various machine learning algorithms such as logistic regression, SVM model, Decision tree, random forest and XG Boost and finally will provide evidence of which model I believe the real estate company should use to gain maximum value.

Data cleaning

The data set contains information about 22651 recent real estate properties sale and 26 attributes such as Acreage, foundation type, finished area, year built, full bath, bedrooms, sale price compared to value, etc. The data set has both continuous and categorical type of variables. The first is to clean the column names, so using snake case formatting I've converted all the column names into lowercase format with spaces replaced by underscore. Next, I dropped the columns which are not relevant to the analysis in the project or does not have enough information to be used in a predictive model such as 'Unnamed: 0', 'Suite/ Condo #', 'property_address', 'parcel_id', 'state', 'legal_reference', 'sale_date'. I dropped all the rows containing null values as the total count of these rows is less than 0.5% of the dataset so it's not going to impact our machine learning model if we ignore these rows. Next, I dropped the duplicates from the dataset. I checked for outlier in the dataset but here isn't enough evidence to drop these values.

Exploratory data Analysis



Correlation matrix is used to understand the relationship between the variables in the data and to check the multicollinearity between the variables. As displayed in diagram here the range is from -1 to 1 and lighter the color of the box higher the relationship between variables and vice versa. A positive value shows a direct positive relation in the variables and vice versa. As we can clearly see from the correlation plot the highest positive correlation is between variables finished area and the building values at 0.85 and there is a negative relation between land value and the year build variables. Most of the properties in the data set are built in years 1950 to 1970 as displayed in the histograms and box plot in appendix. The line charts show how prices varies for properties build in different years.

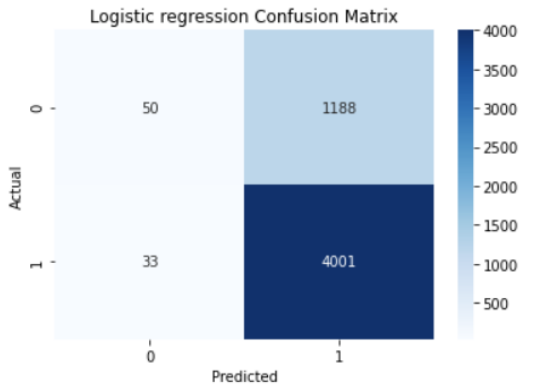
Logistic regression model

Logistic regression model is used to in classification problems i.e., it can take only two values 0 or 1. I'll be using a logistic regression model to predict whether a property sale price value is under or over (1 or 0) than the actual price of the property. One of the major advantage of using the logistic regression model in classification problems is that it provides a very self-explanatory detailed output for each variable in the dataset and it is easier for the user to figure out the significant variables. I'll be using statsmodel.api library in python to generate the summary of logistic model.

Now as the data set contains both categorical and numerical values, I'll first convert all categorical columns into binary format as the logistic regression model only accepts binary or numerical data. For columns Which had data in yes or no format I've replaced these values with 1 and 0 respectively. Using one hot encoding I've converted columns 'land_use', 'property_city', 'city', 'tax_district', 'foundation_type', 'exterior_wall', 'grade' into binary values by creating new columns and dropped the first one among them to avoid the multicollinearity. Next I converted the under and over values in sale_price_compared_to_value column to 1 and 0 format and changed the data type of each variable to numerical format.

The target variable for the logistic model is sale_price_compared_to_value and treating rest of the variables as independent dependent I've divided the data set into train and test in 75:25 ratio using train test split from sklearn library. Taking confidence interval as 95% we can say that the variables having p value less than 0.05 in logistic model output are significant to the model. The most significant variables as per p value analysis are sold_as_vacant, land_value, building_value, land_use_SINGLE FAMILY, tax district city and grade. The coefficient of these variables defines what kind of impact they'll have on the target variable. Sold as vacant has the highest coefficient among the significant variables and has a negative coefficient of -3.04. Next using VIF score I checked for the collinearity in these variables.

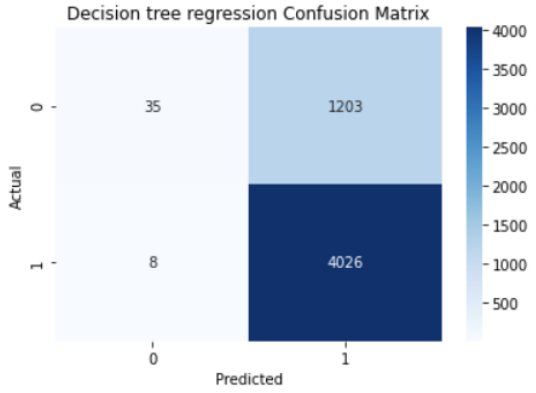
The accuracy of the logistic regression model is 76.8% with a precision of 77% and recall of 99%. The confusion matrix diagram shows the performance of model. Using a cutoff value of 0.5, I transformed the predicted values of the model to binary values to generate the confusion matrix, where anything above 0.5 is interpreted as a property being overvalued and anything below 0.5 as being undervalued. The true positve count is 4001, true negative is 50, false positve is 1188 and false negative is 33. As we can clearly see from the diagram the model correctly predicted 4001 values as 1 which are actually 1 i.e., overvalued property. For 1188 properties the model predicted them to be overvalued but in reality they were not.



Similerly we can see the model predicted 50 properties that are actually undervalued. The real estate company should target these properties to generate more revenue. The true negative values are the one where we predicted a property to be undervalues and it was actually overvalues so this will lead to a loss for the real estate company. Finally I can say that this model is not that convincing because of the higher true negative count

Decision Tree Model

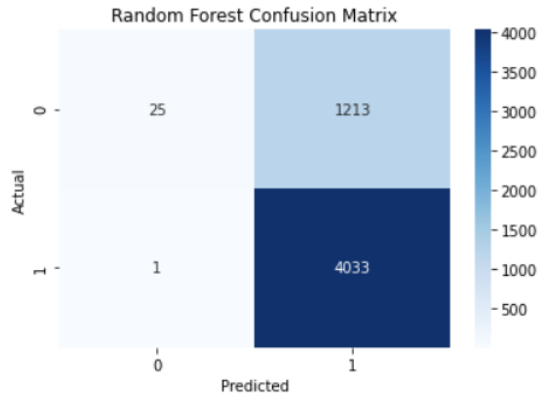
Now I'll be creating a decision tree models with the same set of variables used in the logistic model and will try to comapre the results of botht the models. Decision tree models are mostly used in customer segametation problem and customer churn predictions as they are very easy to understand. Since the business problem of this project is a classification problem I'll use DecisionTreeClassifier from sklearn library to build the model. The accuracy of the decision tree model is 77%. The presision of the model is 77% and the recall is 99%. The appendix shows the decision tree model plot, I've taken the max depth as 4 and minimum sample leaf as 5. limits the depth of the tree and can help prevent overfitting by constraining the complexity of the tree. Increasing max_depth allows the model to learn more complex interactions between features, but can also lead to overfitting if set too high. The accuracy, precision and recall are almost the same as that of the logistic regression model.



The confusion matrix on the left hand side shows the decision tree model performance. The confusion matrix values are also very similer to that of the logistic regression confusion matrix. The true positve count is 4026, true negative is 35, false positve is 1203 and false negative is 8. As dicussed in the logistic regression model, the minimun the count of false negative the better the model. In this decision tree model the chances of loss are very less as comapred to the logistic regression model. The most important features of the decision tree model is dispalyed using a bar plot in the appendix.

Random Forest model

Random forest model similar to decision tree can be used for both classification and regression problems. Using RandomForestClassifier from sklearn library I've created a random forest model. The accuracy of the random forest model is 77%. The presision of the model is 77% and the recall is 99%. The max depth for model is 4 along with 100 n_estimators.



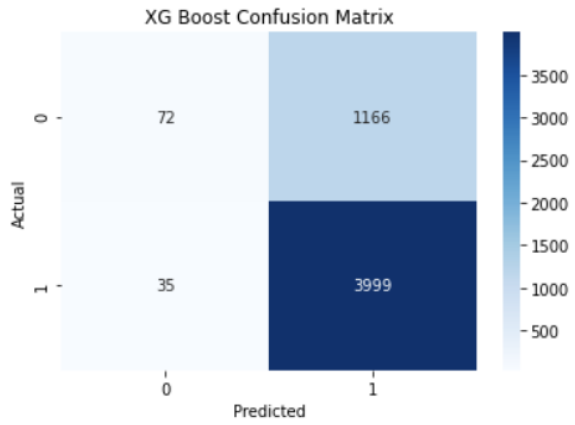
The performance of the random forest model is shown by the confusion matrix. The true positve count is 4033, true negative is 25, false positve is 1213 and false negative is 1.

The false negative count of this model is 1 meaning that there is only 1 prediction where the investment of the real estate company will result in loss, as the property we predicted as undervalued is actually overvalued.

The accuracy, precision and recall are almost the same for logistic regression, decision tree and random forest models. Based on the confuion matrix statistics I can say that the random forest model is the safest model out of all three models created so far as the false negative count is least for this model. The chances of the company losing money are very low.

XG Boost

Similar to the decision tree and random forest, XG boost can also be applied to both classification and regression problems. I'll use the XGBClassifier for creating a gradient boost model. This machine learning algorithm is mostly used to improve the existing decision tree models. I'll use the same train and test split data to create this model. The accuracy of the model is 77%, precision is 77% with a recall of 99%. As accuracy, precision and recall score are highest for XG boost model among all the models. The below confusion matric shows the performance of the XG boost model.



The true positve count is 3999, true negative is 72, false positve is 1166 and false negative is 35.

The number of correct prediction in this model is heighest i.e., 4071 also the 35 false negatives count of this model is heighest among all models meaning that there are 35 predictions where the investment of the real estate company will result in loss, as the property we predicted as undervalued is actually overvalued.

This means that xg boost model is not the safest as compared to other models even though it's accuracy and preciosn is maximum of all the models.

Model Comparison

The below table compares the accuracy, precision, recall, false negative and true negative values. I've included false negative and true negative count for model comparison as the accuracy, precision and recall alone are not enough to determine which model is the best performing for the company. As mentioned in the introduction section, the main objective of the project is to build a model that can be used by the real estate company to target those properties which are undervalued to get the best value deal.

	Logistic Regression	Decision Tree	Random Forest	XG Boost
Accuracy	76.8	77.02	76.9	77.2
Precision	77.1	76.99	76.8	77.4
Recall	99.1	99.8	99.9	99.1
False Negatives	33	8	1	35
True Negatives	50	35	25	72

The accuracy and the precision are highest for the XG boost model and the recall is maximum for random forest model. True negatives are undervalued properties which the models predicted correctly

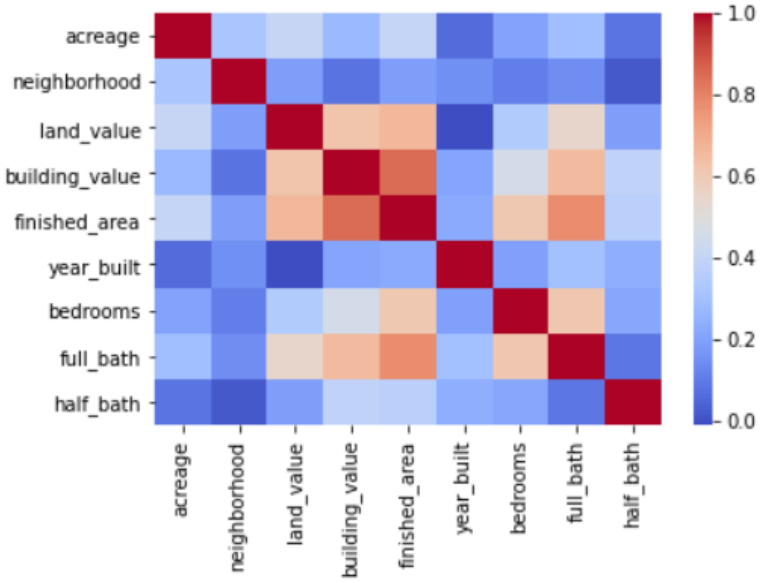
and should be targeted by the company. False negative values are the values which the model predicted the property to be undervalued but the actual price was over the predicted price, this will result in loss to the company. To get the best value deal we should have a model with higher true negative count and a lower false negative count. Random forest has the least false negative and true negative values out of all. I would recommend the company to use this model as the chances of making a bad investment are very low as compared to other models.

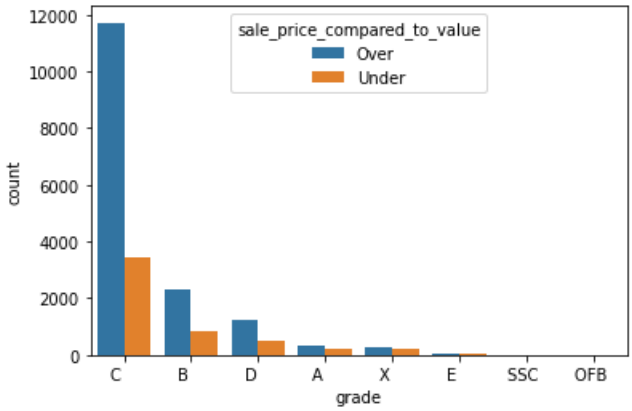
Appendix

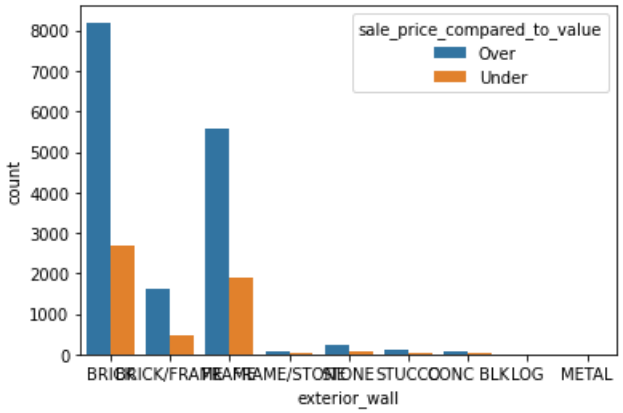
22,651 rows and 26 columns

Show [All columns] Update

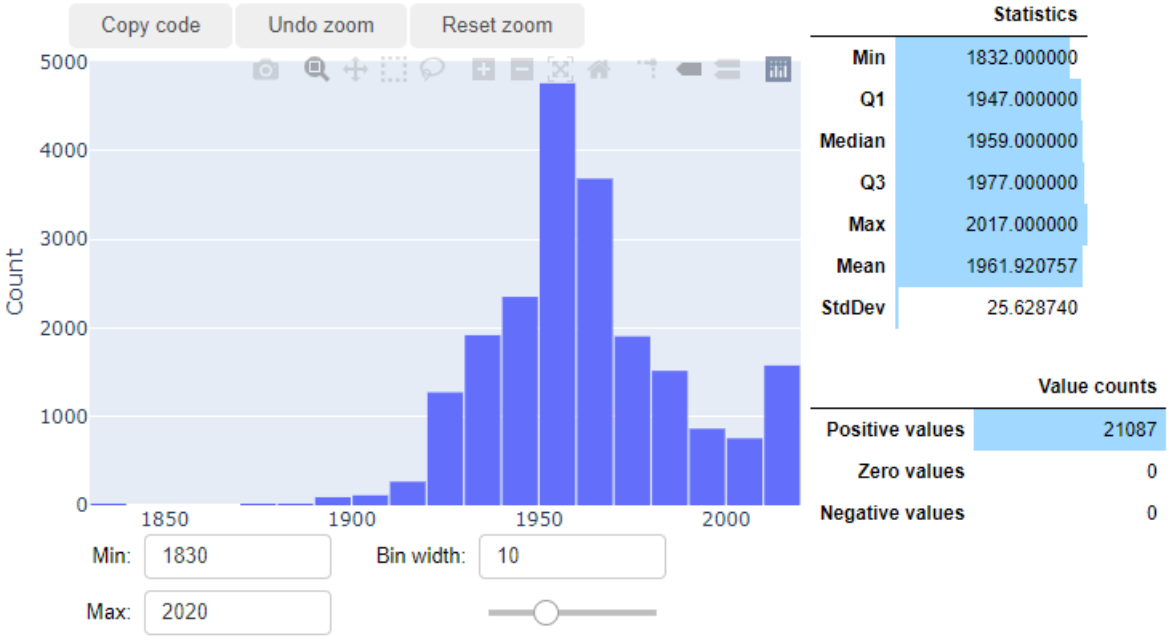
Column	Data type	Unique values	Missing values
Unnamed: 0	int64	22,651	0
Parcel ID	object	19,720	0
Land Use	object	4	0
Property Address	object	20,449	2 - 0.0%
Suite/ Condo #	float64	0	22,651 - 100.0%
Property City	object	11	2 - 0.0%
Sale Date	object	1,044	0
Legal Reference	object	22,452	0
Sold As Vacant	object	2	0
Multiple Parcels Involved in Sale	object	2	0
City	object	10	0
State	object	1	0
Acreage	float64	407	0
Tax District	object	7	0
Neighborhood	int64	189	0
Land Value	int64	886	0
Building Value	int64	4,298	0
Finished Area	float64	5,955	1 - 0.0%
Foundation Type	object	7	1 - 0.0%
Year Built	int64	125	0
Exterior Wall	object	9	0
Grade	object	10	0
Bedrooms	float64	12	3 - 0.0%
Full Bath	float64	11	1 - 0.0%
Half Bath	float64	4	108 - 0.5%
Sale Price Compared To Value	object	2	0


acreage
neighborhood
land_value
building_value
finished_area
year_built
bedrooms
full_bath
half_bath
1.0
0.8
0.6
0.4
0.2
0.0


sale_price_compared_to_value
Over
Under
count
grade
C
B
D
A
X
E
SSC
OFB

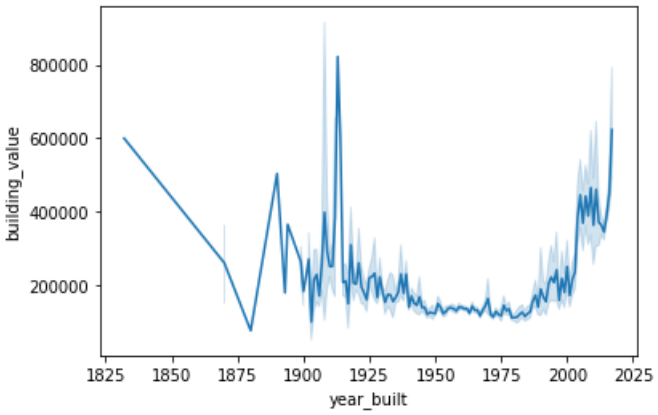

sale_price_compared_to_value
Over
Under
count
exterior_wall
METAL

Histogram



Statistics	
Min	1832.000000
Q1	1947.000000
Median	1959.000000
Q3	1977.000000
Max	2017.000000
Mean	1961.920757
StdDev	25.628740

Value counts	
Positive values	21087
Zero values	0
Negative values	0

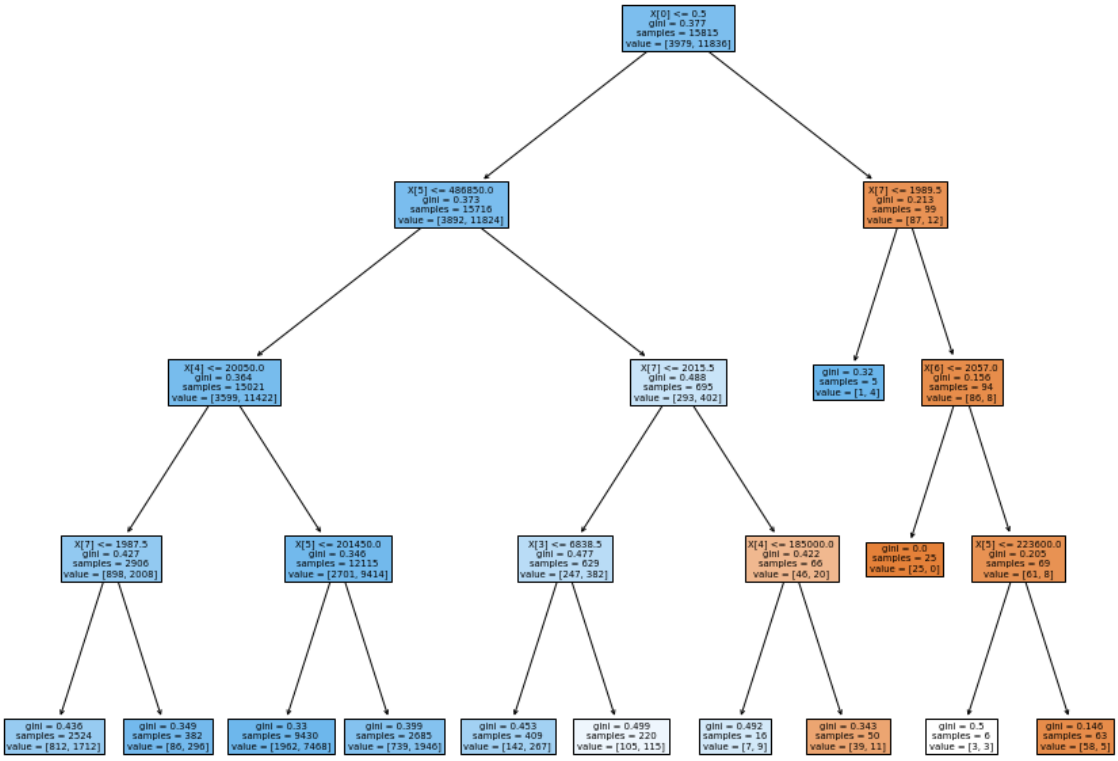


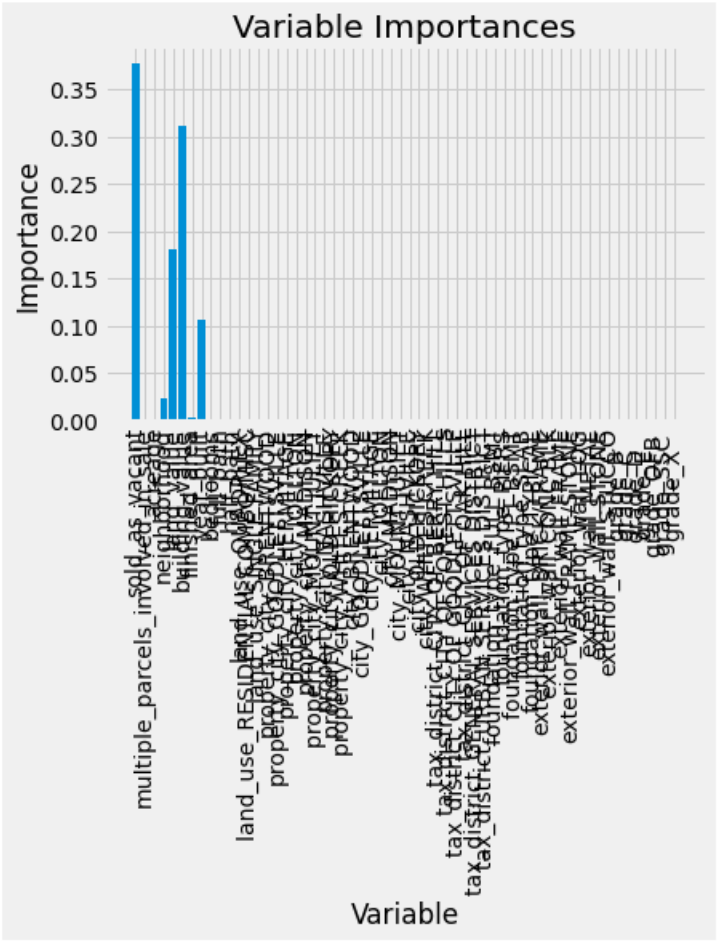
Logit Regression Results

Dep. Variable:	sale_price_compared_to_value	No. Observations:	15815
Model:	Logit	Df Residuals:	15764
Method:	MLE	Df Model:	50
Date:	Sun, 19 Mar 2023	Pseudo R-squ.:	0.03222
Time:	19:33:38	Log-Likelihood:	-8633.5
converged:	False	LL-Null:	-8921.0
Covariance Type:	nonrobust	LLR p-value:	2.628e-90

	coef	std err	z	P>|z|	[0.025	0.975]
const	-3.2927	1.263	-2.607	0.009	-5.768	-0.817
sold_as_vacant	-3.0459	0.210	-14.511	0.000	-3.457	-2.634
multiple_parcels_involved_in_sale	0.4546	0.424	1.072	0.284	-0.377	1.286
acreage	-0.0551	0.053	-1.039	0.299	-0.159	0.049
neighborhood	6.187e-06	1.25e-05	0.495	0.621	-1.83e-05	3.07e-05
land_value	2.509e-06	3.38e-07	7.428	0.000	1.85e-06	3.17e-06
building_value	-1.079e-06	2.17e-07	-4.969	0.000	-1.5e-06	-6.53e-07
finished_area	-5.347e-05	4.99e-05	-1.071	0.284	-0.000	4.44e-05
year_built	0.0015	nan	nan	nan	nan	nan
bedrooms	0.0016	0.030	0.054	0.957	-0.057	0.060
full_bath	-0.0226	0.036	-0.621	0.534	-0.094	0.049
half_bath	0.0120	0.047	0.255	0.799	-0.080	0.104
land_use_QUADPLEX	-0.1356	0.422	-0.321	0.748	-0.964	0.692
land_use_RESIDENTIAL COMBO/MISC	-1.2178	0.425	-2.864	0.004	-2.051	-0.384
land_use_SINGLE FAMILY	0.1788	0.090	1.994	0.046	0.003	0.355
property_city_BRENTWOOD	7.5281	1.67e+06	4.52e-06	1.000	-3.27e+06	3.27e+06
property_city_GOODLETTSVILLE	-0.4480	1.21e+07	-3.71e-08	1.000	-2.37e+07	2.37e+07
property_city_HERMITAGE	-0.2166	nan	nan	nan	nan	nan
property_city_JOELTON	-0.8049	nan	nan	nan	nan	nan
property_city_MADISON	-0.6842	6.04e+06	-1.13e-07	1.000	-1.18e+07	1.18e+07
property_city_MOUNT JULIET	-0.2209	nan	nan	nan	nan	nan
property_city_NASHVILLE	-7.6833	nan	nan	nan	nan	nan
property_city_OLD HICKORY	-0.4764	7.25e+09	-6.57e-11	1.000	-1.42e+10	1.42e+10
property_city_WHITES CREEK	-0.4545	4.83e+07	-9.41e-09	1.000	-9.47e+07	9.47e+07
city_BRENTWOOD	-7.5160	4.9e+05	-1.53e-05	1.000	-9.61e+05	9.61e+05
city_GOODLETTSVILLE	-0.4480	1.21e+07	-3.71e-08	1.000	-2.37e+07	2.37e+07
city_HERMITAGE	-0.2166	nan	nan	nan	nan	nan
city_JOELTON	-0.8049	nan	nan	nan	nan	nan
city_MADISON	-0.6842	6.04e+06	-1.13e-07	1.000	-1.18e+07	1.18e+07
city_MOUNT JULIET	-0.2209	nan	nan	nan	nan	nan
city_NASHVILLE	7.3608	nan	nan	nan	nan	nan
city_OLD HICKORY	-0.4764	2.52e+09	-1.89e-10	1.000	-4.94e+09	4.94e+09
city_WHITES CREEK	-0.4545	4.83e+07	-9.41e-09	1.000	-9.47e+07	9.47e+07
tax_district_CITY OF BERRY HILL	2.9273	1.026	2.853	0.004	0.916	4.939
tax_district_CITY OF FOREST HILLS	0.7644	0.099	7.753	0.000	0.571	0.958
tax_district_CITY OF GOODLETTSVILLE	1.7591	0.308	5.718	0.000	1.156	2.362
tax_district_CITY OF OAK HILL	1.0682	0.143	7.470	0.000	0.788	1.348
tax_district_GENERAL SERVICES DISTRICT	1.7844	0.073	24.424	0.000	1.641	1.928
tax_district_URBAN SERVICES DISTRICT	1.4524	nan	nan	nan	nan	nan
foundation_type_FULL BSMT	-0.0560	0.037	-1.496	0.135	-0.129	0.017
foundation_type_PIERS	0.2997	0.487	0.615	0.539	-0.656	1.255
foundation_type_PT BSMT	0.0324	0.060	0.544	0.586	-0.084	0.149
foundation_type_SLAB	0.0768	0.086	0.890	0.374	-0.092	0.246
foundation_type_TYPICAL	-0.4941	nan	nan	nan	nan	nan
exterior_wall_BRICK/FRAME	0.0820	0.069	1.184	0.236	-0.054	0.218
exterior_wall_CONC BLK	-0.2416	0.267	-0.903	0.366	-0.766	0.283
exterior_wall_FRAME	0.0767	0.043	1.767	0.077	-0.008	0.162
exterior_wall_FRAME/STONE	0.1426	0.271	0.526	0.599	-0.389	0.674
exterior_wall_LOG	-0.1322	0.686	-0.193	0.847	-1.477	1.213
exterior_wall_METAL	0.4037	1.127	0.358	0.720	-1.805	2.612
exterior_wall_STONE	0.0762	0.156	0.489	0.625	-0.229	0.382
exterior_wall_STUCCO	0.2880	0.220	1.311	0.190	-0.142	0.719
grade_B	0.1937	nan	nan	nan	nan	nan
grade_C	0.3629	0.117	3.113	0.002	0.134	0.591

grade_D	-0.0104	0.128	-0.082	0.935	-0.261	0.240
grade_E	-1.0508	0.207	-5.079	0.000	-1.456	-0.645
grade_OFB	16.1305	nan	nan	nan	nan	nan
grade_SSC	-16.6246	nan	nan	nan	nan	nan
grade_X	0.1962	0.178	1.104	0.270	-0.152	0.544




Variable Importances
Importance
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Variable