# Exploring the Impact of Property Characteristics on Tax Classifications In the Boroughs of New York City

Padma Sree Manda, Hari Charan Reddy Karra, Mehran Doveisimogoei, and Sameer Shaik

Department of ITDS, University of North Texas

DSCI 5260: Business Process Analytics

Sameh Shamroukh

05/02/2025

## Table of Contents

## List of Tables

## List of Figures

**Acknowledgements**

We would like to extend our deepest gratitude to everyone who played a role in the successful completion of this project. First and foremost, we express our sincere appreciation to our faculty advisor, Sameh Shamroukh, for the exceptional guidance, timely feedback, and unwavering encouragement throughout our research journey. Their expertise was instrumental in helping us overcome significant challenges and refining our approach.

We are also profoundly thankful to the Department of Business Analytics at the University of North Texas. The resources, knowledge, and supportive academic environment provided were vital to our ability to conduct this work effectively. The backing of our instructors, combined with access to institutional resources, significantly bolstered our research efforts.

Lastly, we recognize the incredible support and collaboration within our team. Each member brought unique contributions, and our collective effort was essential in achieving the successful outcome of this project. Together, we turned our vision into reality.

**Abstract**

This research aims to develop a predictive model for property tax classification in New York City. It focuses on accurately predicting the tax class of properties based on attributes such as property use, zoning, and building characteristics. Accurate property tax classification is crucial for ensuring equitable tax burdens and influencing decisions made by property owners, real estate developers, and policymakers. The dataset, sourced from Manhattan's property assessments, includes a mix of numerical and categorical features, requiring advanced machine learning techniques for effective classification.

To identify the most important predictors, the research employs the Lasso model for feature selection, helping to highlight the most influential variables in predicting property tax classes. Various classification models, including Decision Trees, Random Forests, and Logistic Regression, were explored. Logistic Regression was ultimately selected due to its superior performance on the imbalanced dataset, demonstrating better precision and recall, especially for the smaller tax classes. The findings suggest that while machine learning models can provide more accurate predictions compared to traditional methods, Logistic Regression’s performance makes it particularly suited for real-world applications, especially when features have high correlations and when dealing with class imbalance.

**Keywords**: Property tax classification, *Machine learning, Lasso,* Logistic Regression

# Introduction

Property tax classification serves as a cornerstone of fiscal policy, profoundly impacting taxation rates and financial planning for stakeholders in the real estate sector. Accurate classification ensures equitable tax distribution and influences decisions made by property owners, developers, and policymakers.

However, traditional methods of property tax assessment often face challenges related to accuracy and fairness, leading to potential disparities in tax burdens. As cities and local governments rely on property taxes as a primary revenue source, accurate and fair classification is crucial for maintaining equitable tax burdens across different property types.

## Problem Statement

This study aims to develop a predictive model for classifying properties into their Current Tax Class by analyzing various property attributes. These attributes include Property Use, Number of Units, Zoning and Occupancy. By leveraging historical property assessment data, particularly from New York City (NYC), the model seeks to identify primary predictors that influence tax classification. The classification of properties into appropriate tax classes is essential for maintaining a balanced and transparent taxation system.

Misclassification can lead to revenue inefficiencies, taxpayer disputes, and inequitable tax burdens. While past studies have examined property tax assessments from valuation perspectives, there remains a research gap in predictive modeling techniques that optimize classification accuracy. This study addresses this gap by employing machine learning and statistical methods to enhance tax class prediction models, providing a more data-driven approach to property taxation.

Developing an accurate predictive model for property tax classification is critical for improving assessment accuracy, promoting fairness, informing policy decisions, and assisting

stakeholders. The insights generated from predictive models can aid policymakers in refining tax policies to enhance fairness and efficiency in revenue generation. Furthermore, real estate developers and property owners can leverage these insights to better understand potential tax liabilities, improving financial planning and investment strategies.

**Objective**

The primary objective of this research is to develop a robust predictive model that accurately classifies properties into their respective tax classes using historical assessment data. The model aims to:

- Identify Key Predictors: Determine the most influential property attributes that affect tax classification.
- Enhance Predictive Accuracy: Utilize advanced machine learning techniques to improve the precision of tax class predictions.
- Facilitate Policy and Planning: Provide insights that support equitable taxation policies and assist stakeholders in anticipating tax obligations.

To conduct this research, we will use a combination of tools for data processing, statistical analysis, and machine learning. Python, with libraries like Pandas and Scikit-learn, will facilitate exploratory data analysis (EDA) and model development. We will use Tableau to visualize the data and generate insights. Machine learning techniques such as Decision Trees, Random Forests, Gradient Boosting, and Neural Networks will help us build classification models, while Logistic Regression and Support Vector Machines will serve as benchmarks. We will apply Eigenvector Spatial Filtering (ESF) to account for spatial autocorrelation that might exist between neighboring properties. We will also apply feature selection methods, such as Recursive Feature Elimination

(RFE), to identify key predictors and explain how each feature contributes to the prediction. PDPs (Partial Dependence Plots) will help us make the decision process more transparent by visualizing the effect of a feature on predictions.

**Research Questions**

Building on the objective of developing an accurate predictive model for property tax classification, this research aims to identify the most influential factors that determine tax class assignments and understand how exemptions affect the distribution of taxable properties. The primary research questions for this project are:

1. What are the most significant property characteristics influencing the classification of a property into different tax classes?
2. How do exemptions (e.g., CURACTEXTOT, CURTRNEXTOT, CURTXBEXTOT) impact the distribution of taxable properties across different classes?

## Literature Review

**Summary**

As a part of our study, we have chosen five articles that are related to our research and are helpful in providing guidance on the steps to move forward in the research. Below is the comprehensive view of the papers based on our understanding of them.

***Machine Learning Approach for Taxation Analysis using Classification Techniques***

A sample of 365 live records is used to examine 24 attributes of income and tax returns for single taxpayers and businesses. Various types of algorithms ranging from Naive Bayes, Decision Tree (J48), Logistic Regression, SVM (SMO), Radial Basis Function Networks, meta-classifiers (Bagging and Boosting) up to the rule-based methods are discussed. The models are validated

using the WEKA data mining tool, and classification accuracy is used as the performance measure (Radha & Lakshmi, 2011) (Gloudemans & Sanderson, 2021).

The research deals with two main aspects: Classifier Selection, the purpose of which is to find the most effective model, and Classifier Fusion, which is the process of integrating multiple classifiers to attain better results (Radha & Lakshmi, 2011). The study supports the idea that Classifier Fusion is usually better than Classifier Selection because it is based on a combination of several data models which boost prediction accuracy in turn. Notably, Decision Trees (J48) and Naïve Bayes hold a better position compared to other algorithms as they can effectively handle big datasets.

Moreover, the importance of the selection feature was pointed out to do or play, because the removal of redundant attributes has been shown to bring about positive improvement in prediction accuracy. This study demonstrates how machine learning techniques can enhance traditional audit selection methods, thus allowing more effective tax compliance monitoring.

Through the analysis of a wide array of classification models used in the framework of tax audit selection, this study furthers the current body of knowledge in the field of data mining as it intersects with financial and tax administration. The findings obtained from this study act as an essential guide toward optimization of the audit operations using machine learning approaches, which among other things involve a thorough examination of the different classification algorithms and their comparative performance metrics.

***The Potential of Artificial Intelligence in Property Assessment***

This journal article explores the use of Artificial Intelligence in property tax assessments and provides a framework for developing a standard application. The authors try to demonstrate

how AI can improve accuracy, objectivity, and efficiency in property valuation compared to traditional methods like Multiple Regression Analysis.

This journal highlights algorithms such as Regression, Classification, Random Forests and Neural Networks produce more accurate results than traditional mass appraisal tools, i.e., tools that are used to evaluate property such as CAMA systems (NTPTS, 2024). The study also provides a detailed summary of common AI techniques and their use in property assessments. The study explains that most of the companies or agencies are leaning towards either Gradient Boosting Methods or Neural Networks as they provide more accurate results compared to other AI methods (Gloudemans & Sanderson, 2021).

The study is both conceptual and exploratory analysis which reviews existing AI methods and their applications in both the private and public sectors of property valuation. It includes case studies such as the Property Valuation Services Corporation (PVSC) of Nova Scotia, Canada and briefly talks about the Wake County in North Carolina and their contract with a software company to interpret the practical use of AI in property assessments (Gloudemans & Sanderson, 2021; Lee, 2022).

***Training and Interpreting Machine Learning Models: Application in Property Tax Assessment***

Machine learning has shown great potential in property valuation, yet its adoption remains slow due to infrastructure limitations and the complexity of its algorithms, which hinder interpretability (Lee, 2022). Property tax assessment, which demands both accuracy and transparency, could benefit significantly from interpretable machine learning models. This study aims to improve the adoption of machine learning in property tax assessment by enhancing model interpretability. A neural network estimates house prices in two provinces, with partial dependence

plots (PDPs) analyzing key variables to make the decision process more transparent. The study promotes explainable AI to improve accuracy, transparency, and stakeholder trust.

This study reveals that machine learning models, particularly neural networks, can enhance property tax assessment accuracy while maintaining interpretability through partial dependence plots (PDPs). Findings show that the site area is the main price determinant in an urban region, while building area is more influential in a rural area, highlighting the need for region-specific valuation models (Lee, 2022). By incorporating interpretability tools, the study addresses barriers to machine learning adoption and supports localized pricing models.

A neural network model is trained using key variables like site area, building area, zoning, road width, and property age. The model, optimized with Adam and ReLU, uses mean squared error (MSE) as the loss function. PDPs enhance interpretability by revealing regional differences in price determinants.

***Prediction Accuracy for Property Tax Mass Appraisal***

This study compares traditional multiple regression analysis with advanced machine learning methods like ridge, lasso, elastic net and geospatial approaches such as eigenvector spatial filtering for property tax mass appraisal. The goal of this study is to evaluate predictive accuracy and explainability of these models in assessing property tax.

Traditional methods like multiple regression analysis doesn't capture complex relationships like spatial dependencies (McCord, Davis, Bidanset, & Hermans, 2022). The study finds that both Machine Learning (ML) models and Eigenvector Spatial Filtering (ESF) approaches have outperformed traditional methods in terms of prediction accuracy, but ML models lack explainability and are difficult to interpret and defend why the property was valued in a certain way. ESF technique is a better alternative that reduces spatial dependency and has better tradeoff

between accuracy and transparency. ESF is easy to interpret and can provide clear reasons for how the property value was determined.

This study uses real estate transaction data from Belfast, Northern Ireland, and Bloomingdale, Illinois to evaluate different models like multiple regression analysis, ESF, ridge, lasso, and elastic net for property evaluation used for tax purposes. These models were trained using cross-validation to tune hyperparameters and are evaluated using different metrics like Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Price-Related Differential (PRD), Coefficient of Dispersion (COD) with an 80/20 train-test split.

***Measuring the Impact of Taxes and Public Services on Property Values: A Double Machine Learning Approach***

This study examines how property values respond to changes in local taxes and public services by using a double machine learning approach. Previous studies missed considering many details like quality of local public services like education and various other factors that affect property prices. This study uses a dataset that consist of 947 local public service controls in Sweden to measure the effect of taxes on property values without letting other factors skew the results and this study also tests the Tiebout hypothesis.

Public service inputs like higher school spending per pupil and high crime rates are decreasing the property values whereas public service outputs like better school quality are increasing it (Grodecka-Messi & Hull, 2022). Areas having more municipalities have greater drop in property prices because of high taxes especially in urban areas, supporting Tiebout hypothesis that says people will move to better tax-service tradeoff areas. A one standard deviation increment in local income is reducing property prices by 0.26 standard deviations where the effect in stronger

in urban areas and rural areas has almost no effect. Double machine learning (DML) with deep-wide neural network has given more accurate and unbiased estimates of tax capitalization compared to traditional methods.

This study uses huge dataset covering 290 municipalities in Sweden over a period of seven years (2010-2016) that consist of 947 different variables covering various factors related to housing, public services and migration. A double machine learning estimator was applied to remove the biases by predicting taxes and property prices separately and a deep wide neural network is used to capture nonlinear relationships between housing prices, taxes, and public services. Further, Monte Carlo simulations are used to compare DML to simpler methods like OLS regression, showing DML produced more accurate and unbiased results (Grodecka-Messi & Hull, 2022).

**Research Gap**

- Existing research is at national or state level but lack granular level of research to understand the ground level analysis. Hence, our research aims to train different models to understand the patterns of property tax at borough level given the differences existing in the policies of taxes based on state or county.
- The studies of these papers focused on various interpretations, mainly accuracy and predicting capabilities. However, it lacked focus on various factors that might be responsible for the higher accuracy as machine learning algorithms are generally “black box” in nature. These models rarely provide transparency that is understandable to non-technical persons. Hence, our research aims to understand the underlying factors and reasons for the classification of property tax classes along with a good predicting capability.

- As per our findings from these papers, regularization techniques such as lasso, ridge and elastic net are rarely used though they are a middle ground for good accuracy and interpretation ability when compared to other machine learning models. As a part of our analysis, we plan to use these models to select features as they provide detailed analysis of the importance of each feature.
- Studies indicated the presence of spatial dependencies in property valuation which cannot be explained or explored by traditional machine learning regression models. As per the research paper Eigenvector Spatial Filtering is an effective technique in terms of prediction accuracy but it is underexplored for property tax classification. This is a new technique which we came across and are unsure of its practicality in terms of classification. We will explore the usage of this model in our project by taking the research paper as guidance.

## Research Design

### Data Description

For this project we sourced data from the official website of New York public data (NYC Open Data, n.d.)[1]. The dataset is sourced by various city employees such as property assessors, property exemption specialists, ACRIS reporting and the department of building reporting, etc.

The dataset is made up of 139 columns and 6.98 million rows. The dataset consists of almost 139 columns and 6.9 million records which consists of information of all 5 boroughs of New York. Out of these records we are planning to use only the data from Borough 1, which is Manhattan.

[1] For more information, see the NYC Open Data website at https://data.cityofnewyork.us/City-Government/Property-Valuation-and-Assessment-Data-Tax-Classes/8y4t-faws/about_data (Last Accessed 19 March 2025)

***Measures - Variables***

We will focus on 41 independent variables that are essential for analysis and model development. The dependent/target variable is the Property Tax Class (**CURTAXCLASS**), a categorical column which consists of the values of 4 different tax classes of New York City's property assessment guidelines. Each of these 4 classes represents the nature of occupancy of a particular property. Below is the table providing descriptions of each property class (Avenue Law Firm, n.d.)[2].

Table 1: Property Tax Class Description

| Property Class | Description | Example |
|---|---|---|
| **Class 1** | 1 to 3 unit residential property. | Single-family homes |
| **Class 2** | Residential Property with more than 3 units. | Apartment buildings, Condos, etc. |
| **Class 3** | Special franchise property. | Utility infrastructure. |
| **Class 4** | All other real property, like office buildings, factories, hotels and lofts. | Commercial buildings, warehouses, hotels, retail stores |

The dataset includes both numerical and categorical variables. The categorical columns of the dataset mainly include information about the characteristics of the property such as LOT, EASEMENT, BUILT YEAR, BUILDING CLASS, OWNER, SQFT etc. Some of the important numerical variables are listed below in the table.

[2] For more information, see the Avenue Law Firm website at https://www.avenuelawfirm.com/property-taxes-determined-new-york-city/#property-classifications-in-nyc (Last Accessed on 19 March 2025)

Table 2: Few Key Numerical Variables

| S.No | Key Numerical Variables |
|---|---|
| **1** | Current Market Assessed Land Value |
| **2** | Current Actual Assessed Land Value |
| **3** | Current Taxable Assessed Total Value |
| **4** | Current Taxable Exemption Total Value |
| **5** | Number of units |
| **6** | Renovation details |
| **7** | Characteristics of the property such as square footage of the building etc |

## Research Methods

Our analysis focuses on Machine Learning models to understand the patterns and predict the outcomes of feature set. We mainly use Lasso and Random Forest Classifier to get the list of features that are highly relevant to our target, which is Tax Class. Other models such as Logistic Regression, Decision Tree Classifier along with Random Forest Classifier are being used to predict the tax class based on the prepared data. The models used are described below to understand their algorithms and use cases.

### *Lasso Regression*

Lasso is a linear model from sklearn.linear_model module in Python that we used here primarily for feature selection (scikit learn , n.d.)[3]. In datasets with large number of features, like our dataset that contains over 40 variables, Lasso helps in finding the most relevant features

[3] For more information, see the scikit learn documentation at https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html (last accessed 1 May 2025)

automatically. It will do this by selecting the less important variables and reducing their influence by setting their coefficients to zero (Agrawal, 2023)[4].

Lasso's biggest advantage is that it can handle multicollinearity (a problem we identified earlier during correlation testing). It will eliminate the redundant features and only keep those features that are meaningful land important in predicting the property tax classes. This will reduce the risk of overfitting.

Lasso doesn't capture non-linear relationships like decision trees or neural networks, instead it will help in providing valuable insights during the feature selection phase. Lasso will help simplify more complex models and can help gain a better understanding of factors that will greatly influence how tax classes are classified

***Logistic Regression***

Logistic Regression model is used from "sklearn.linear_model" module in Python that will help us develop a classification model in predicting property tax class (scikit learn developers, n.d.)[5]. Logistic regression is a statistical machine learning model that estimates the probability of categorical dependent variable using one or more independent variable. It will be suitable for binary as well as for Multi class classification and hence it can be applicable for predicting tax classes (CURTAXCLASS) that contains multiple categorical values.

Logistic regression is a significant machine learning algorithm that assumes linear relationship between input features and log-odds of target variable. It is based on logistic/sigmoid function that will map the predicted values in between 0 and 1. Then these values are used to

[4] For more information, see the article at
Feature Selection Using Lasso Regression | by Saurav Agrawal | Medium (Last Accessed on 15 April 2025)
[5] For more information, see the scikit learn documentation at
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html (Last Accessed on 25 April 2025)

predict the most likely class. One main advantage of the model is its transparency as it allows us to look at the weights assigned to each feature that will help us better understand which feature is influencing more in predicting property tax class.

However, it doesn't perform well when the relationship between features and the outcome is non-linear or when class distributions are highly imbalanced. To address this, regularization techniques like L1 (Lasso) and L2 (Ridge) are applied that will help us prevent overfitting by reducing less important feature weights. We can also solve class imbalance issues (Tax Class 2 dominates the dataset), by using a weighted classification approach and evaluate performance using metrics like the weighted F1-score.

In summary, Logistic regression is a simple and transparent model that will provide us with valuable insights and builds a strong foundation before moving on to more complex models like decision trees and neural networks

***Decision Tree Classifier***

DecisionTreeClassifier model is used from the "sklearn.tree" package of Python to develop a classification model that predicts property tax class (Scikit Learn, n.d.)[6]. Decision Tree is a traditional Machine Learning model that works irrespective of the nature of relationship between the predictors and the target variable. Unlike other traditional classification ML models, such as, Logistic Regression or Support Vector Machines (SVM), which require the presence of linear relationship.

The Decision Tree is a well-known machine learning model that operates by recursively dividing input data into separate subsets according to the values of its inherent features. At each

[6] For more information, see the Scikit Learn Documentation at https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html (Last Accessed on 19 March 2025)

step, the algorithm determines which features best separate the data into distinct groups, resulting in a tree-like representation that outlines the decision criteria (Baladram, 2024)[7]. One important measure used in the decision tree algorithms is entropy, which assesses the impurity or disorder present within the data subsets as the tree is being built. Entropy measures the degree of uncertainty or randomness associated with a given set of data points and is crucial for calculating information gain.

As the Decision Tree model is a recursive model, it tends to overfit on the data and study the patterns along with the noise in it. To overcome this challenge, early stopping methods are used to tune the parameters in such a way that the model stops dividing before reaching the complete leaf nodes. Early Stopping Methods include limiting the maximum depth of the tree, trimming the tree structure, and setting minimum sample sizes for splits (GeeksforGeeks, 2024)[8]. By streamlining the tree structure and fostering better generalization, these strategies improve the decision tree's ability to provide precise predictions on new data.

As per the initial understanding or analysis of the data, we believe that there might be no linear relationship between the independent and dependent variables. Hence, we plan to work on models that don't have limitations on the type of relationship, such as, Decision Tree Classifier, Random Forest Classifier, Neural Networks, etc., while the linear ML models such as Logistic Regression and Support Vector Machines will work as a benchmark.

---

[7] For more information, see the article at https://medium.com/data-science/decision-tree-classifier-explained-a-visual-guide-with-code-examples-for-beginners-7c863f06a71e (Last Accessed on 19 March 2025)

[8] For more information, see the GeeksforGeeks article at https://www.geeksforgeeks.org/pruning-decision-trees/ (Last Accessed on 19 March 2025)

***Random Forest Classifier***

Random Forest model is used from the "sklearn.ensemble" package in Python that will help us develop a classification model in predicting property tax class (Scikit-learn Developers, n.d.)[9]. Logistic regression is an ensemble learning method that doesn't just rely on a single decision tree that may overfit the training data, it takes the average (majority votes) of several trees and makes the final classification decision.

In random forest, each tree is trained using random sample in the dataset with replacement using bagging techniques and at each split within the tree, random subsets of features are used. This randomness will introduce diversity and will help the model to generalize well for unseen data. So, random forest performs well even if the relationships between the features and target variables are complex. Also, it is robust to outliers and can work with both categorical and numerical features without any extensive preprocessing.

To avoid overfitting and to improve model's performance, we can adjust key parameters like number of trees(n_estimators), maximum depth (max_deptth) and min_samples_split. These parameters are fine tuned to restrict overfitting to the train data and increase the model's performance on test data.

For our high dimensional dataset that contains many features, strong correlations, Random Forest will help us build a model that's powerful and reliable for predicting correct tax class for NYC properties

[9] For more information, see the scikit learn documentation at https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (Last access on 25 April 2025)

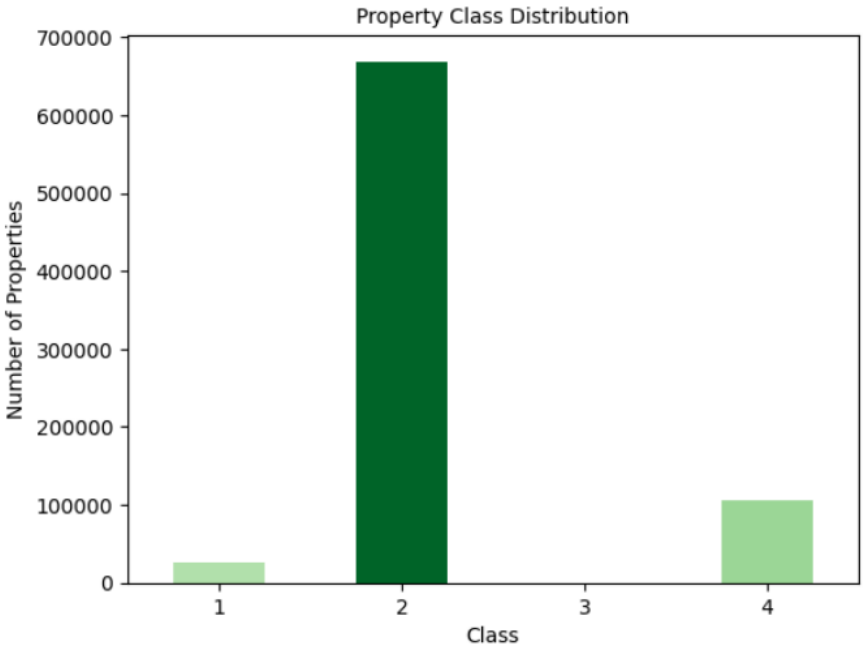
## Data Analysis

As a part of our initial analysis, we have checked for duplicate values and missing values in the data set. We have found that there are no duplicate values in the data set and there are certain columns with high missing values, some of those columns are categorical variables where missing value means blank which indicates that the corresponding property does not belong to that column.

We have checked the distribution of some of the key numerical variables through various visualizations such as histogram and kdeplot. Based on the plots, it was clear that they were not normally distributed.

We have filtered the dataset based on the YRBUILT feature to remove the buildings that build prior to 1900 which left us with a dataset of nearly 800,000 instances.A bar plot has been designed to display the distribution of property tax classes. From the graph below we can see that the target variable is imbalanced with class 2 having higher weightage than the combined class 1, 3, & 4 weightages.

Figure 1: Property Tax Classes Distribution



As mentioned above, the distribution of properties into the classes is highly imbalanced with Class 2 having nearly 75% weightage. Hence, while evaluating the model performance we want to use weighted techniques or f1-score.

**Exploratory Data Analysis**

As a part of exploratory data analysis, we have examined categorical variables and numerical variables. For categorical variables we have looked at its value count to understand the weightage of their individual elements and to further impute these and transform the categorical variables for model building. As per the computed weightage of categorical variables using value_counts(), we have dropped some columns that have skewed results. The results and code snippets can be referred to in Appendix[10].

For numerical variables we have conducted descriptive analysis, some of these numerical variables influence the target variable that is current property tax class. These numerical variables are Current Market Assessed Land Value, Current Market Assessed Total Value, Current Actual Assessed Land Value, etc.

[10] For more information refer to Appendix which has code snippets of the categorical variables.

Table 3: Exploratory Descriptive Analysis Report

| | CURMKTLAND | CURMKTTOT | CURACTLAND | CURACTTOT | CURACTEXTOT | CURTXBTOT | CURTXBEX TOT |
|---|---|---|---|---|---|---|---|
| **count** | 1048575 | 1048575 | 1048575 | 1048575 | 1048575 | 1048575 | 1048575 |
| **mean** | 783339.7 | 3580026 | 319072.1 | 1516469 | 410904.2 | 1414581 | 409443.3 |
| **std** | 16250682 | 37486226 | 7308653 | 16859750 | 11132940 | 16701524 | 11131093 |
| **min** | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **25%** | 27213 | 224230.5 | 12200 | 100182 | 0 | 91831 | 0 |
| **50%** | 57155 | 408106 | 25484 | 177708 | 0 | 162217 | 0 |
| **75%** | 203000 | 1201000 | 80001 | 456643 | 0 | 377980 | 0 |
| **max** | 6.34E+09 | 9.66E+09 | 2.86E+09 | 4.35E+09 | 3.3E+09 | 4.35E+09 | 3.3E+09 |

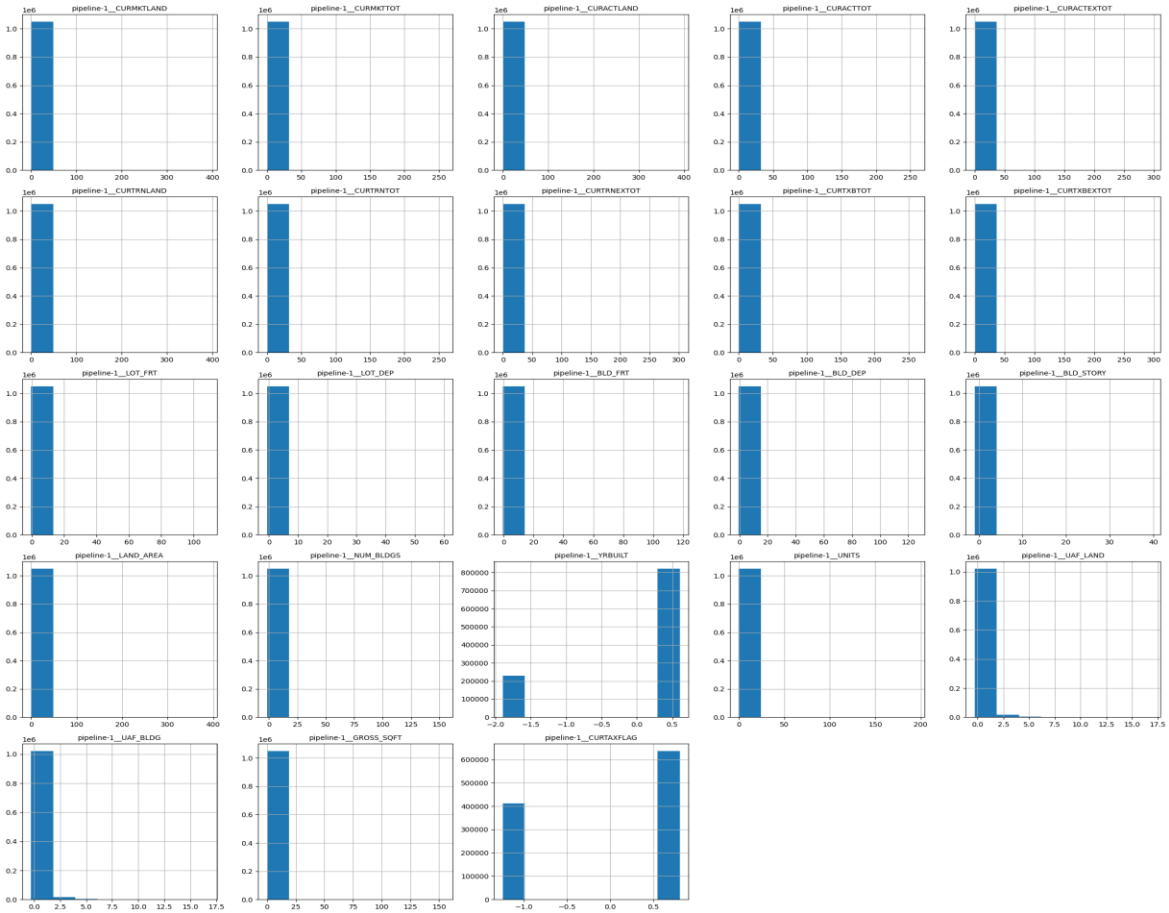
**Assumptions Testing**

To evaluate the appropriateness of linear classification models for our dataset, we conducted multiple exploratory analyses focusing on feature distribution (normality) and inter-feature dependence (correlation).

***Normality Test***

First, histograms of numerical features were plotted to assess the distributional properties of the data. These diagnostics are essential in determining whether linear models, which assume normally distributed inputs and minimal multicollinearity, are suitable for modeling the underlying data structure.

Figure 2: Normality Distribution Test (Histogram)




The histograms reveal that most numerical variables are heavily skewed or exhibit long-tailed distributions, indicating significant deviations from normality. This pattern is especially pronounced in variables related to property valuation and land characteristics, such as CURACTTOT, CURTXBEXTOT, and GROSS_SQFT, where the majority of data points cluster near the lower end of the scale with sparse representation in the upper ranges. Such non-Gaussian distributions violate a fundamental assumption of linear models—that predictors should have a linear and homoscedastic relationship with the response variable.

***Correlation Testing***

Pearson correlation coefficients were computed to examine the degree of linear dependence between features. A Table has been listed below to display the pair of features that are highly correlated.

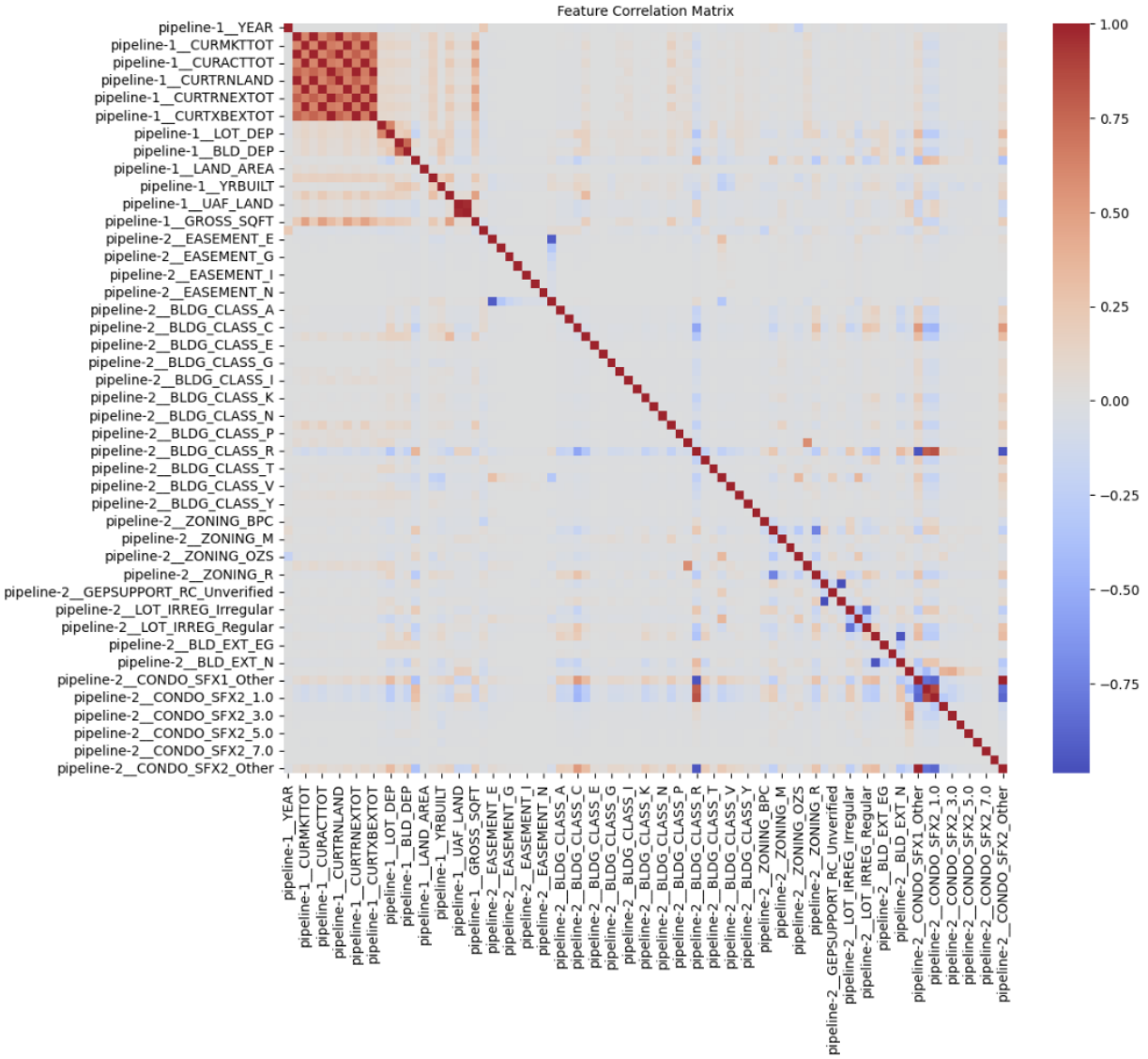
Table 4: List of Feature Pairs that are highly correlated

| | **Feature 1** | **Feature 2** | **Correlation** |
|---|---|---|---|
| **1** | pipeline-2__CONDO_SFX2_Other | pipeline-2__CONDO_SFX1_Other | 1.000 |
| **2** | pipeline-1__CURTXBEXTOT | pipeline-1__CURACTEXTOT | 1.000 |
| **3** | pipeline-1__CURACTLAND | pipeline-1__CURTRNLAND | 1.000 |
| **4** | pipeline-1__CURMKTLAND | pipeline-1__CURACTLAND | 0.999 |
| **5** | pipeline-1__CURACTTOT | pipeline-1__CURTXBTOT | 0.999 |
| **6** | pipeline-1__CURACTTOT | pipeline-1__CURMKTTOT | 0.999 |
| **7** | pipeline-1__CURMKTLAND | pipeline-1__CURTRNLAND | 0.999 |
| **8** | pipeline-1__CURTXBTOT | pipeline-1__CURTRNTOT | 0.999 |
| **9** | pipeline-1__CURMKTTOT | pipeline-1__CURTXBTOT | 0.998 |
| **10** | pipeline-1__CURTRNEXTOT | pipeline-1__CURTXBEXTOT | 0.998 |
| **11** | pipeline-1__CURTRNEXTOT | pipeline-1__CURACTEXTOT | 0.998 |
| **12** | pipeline-1__CURTRNTOT | pipeline-1__CURACTTOT | 0.998 |
| **13** | pipeline-1__CURMKTTOT | pipeline-1__CURTRNTOT | 0.997 |
| **14** | pipeline-1__UAF_BLDG | pipeline-1__UAF_LAND | 0.962 |
| **15** | pipeline-2__CONDO_SFX2_1.0 | pipeline-2__CONDO_SFX1_R | 0.880 |
| **16** | pipeline-2__CONDO_SFX2_1.0 | pipeline-2__BLDG_CLASS_R | 0.830 |

| | | | |
|---|---|---|---|
| **17** | Pipeline-2__LOT_IRREG_Irregular | pipeline-2__LOT_IRREG_Regular | -0.809 |
| **18** | pipeline-2__CONDO_SFX1_R | pipeline-2__CONDO_SFX2_Other | -0.820 |
| **19** | pipeline-2__CONDO_SFX1_Other | pipeline-2__CONDO_SFX1_R | -0.820 |
| **20** | pipeline-2__CONDO_SFX2_1.0 | pipeline-2__CONDO_SFX1_Other | -0.869 |
| **21** | pipeline-2__CONDO_SFX2_Other | pipeline-2__CONDO_SFX2_1.0 | -0.869 |
| **22** | pipeline-2__CONDO_SFX1_Other | pipeline-2__CONDO_SFX2_1.0 | -0.869 |
| **23** | pipeline-2__EASEMENT_No Easement | pipeline-2__EASEMENT_E | -0.920 |
| **24** | pipeline-2__BLD_EXT_N | pipeline-2__BLD_EXT_E | -0.951 |
| **25** | pipeline-2__CONDO_SFX2_Other | pipeline-2__BLDG_CLASS_R | -0.956 |
| **26** | pipeline-2__CONDO_SFX1_Other | pipeline-2__BLDG_CLASS_R | -0.956 |
| **27** | pipeline-2__BLDG_CLASS_R | pipeline-2__CONDO_SFX2_Other | -0.956 |
| **28** | pipeline-2__GEPSUPPORT_RC_Exceptions | pipeline-2__GEPSUPPORT_RC_Verified | -0.987 |

The correlation analysis highlights substantial multicollinearity among many of the features. Several pairs of valuation-related variables, such as CURTXEXTOT and CURACTTOT, or CURRTNTOT and CURMKTLOT, demonstrate correlations approaching or exceeding 0.99. This high degree of redundancy can undermine the stability and interpretability of linear classifiers by inflating variance in the estimated coefficients. In addition, strong correlations among binary categorical encodings (e.g., between one-hot encoded condo subtypes) further reinforce the presence of feature dependence in both numerical and categorical domains.

Figure 3: Correlation Matrix, Heat Map




The correlation matrix helps us understand how different features in our dataset relate to one another. One of the key takeaways is that several features related to property value assessments like CURMKTTOT, CURTRNTOT, CURACTTOT, and CURTXBTOT are strongly correlated with each other. This makes sense because they all represent different ways of measuring a property's value, which are often used to determine its tax class. Since these features are very similar, they may be giving the model overlapping information.

On the other hand, features like zoning codes, building types, and easement categories which were turned into separate columns through one-hot encoding don't show strong correlations with other features. This means they provide unique and useful signals to the model. Overall, the matrix shows that while some value-based features are closely related, the rest of the data offers a good variety of information. This helps us decide whether we want to keep all the features or reduce some of them to make the model simpler and easier to interpret.

Given the combination of non-normal feature distributions and extensive inter-feature correlation, we conclude that the underlying data structure violates the assumptions required for the reliable performance of linear classification models. As a result, non-linear models, such as tree-based ensembles or neural networks, may offer a more robust framework for capturing the complex interactions and non-linear boundaries inherent in the dataset. Therefore, these models do not require strict adherence to normality or independence assumptions and are better suited for this classification task.

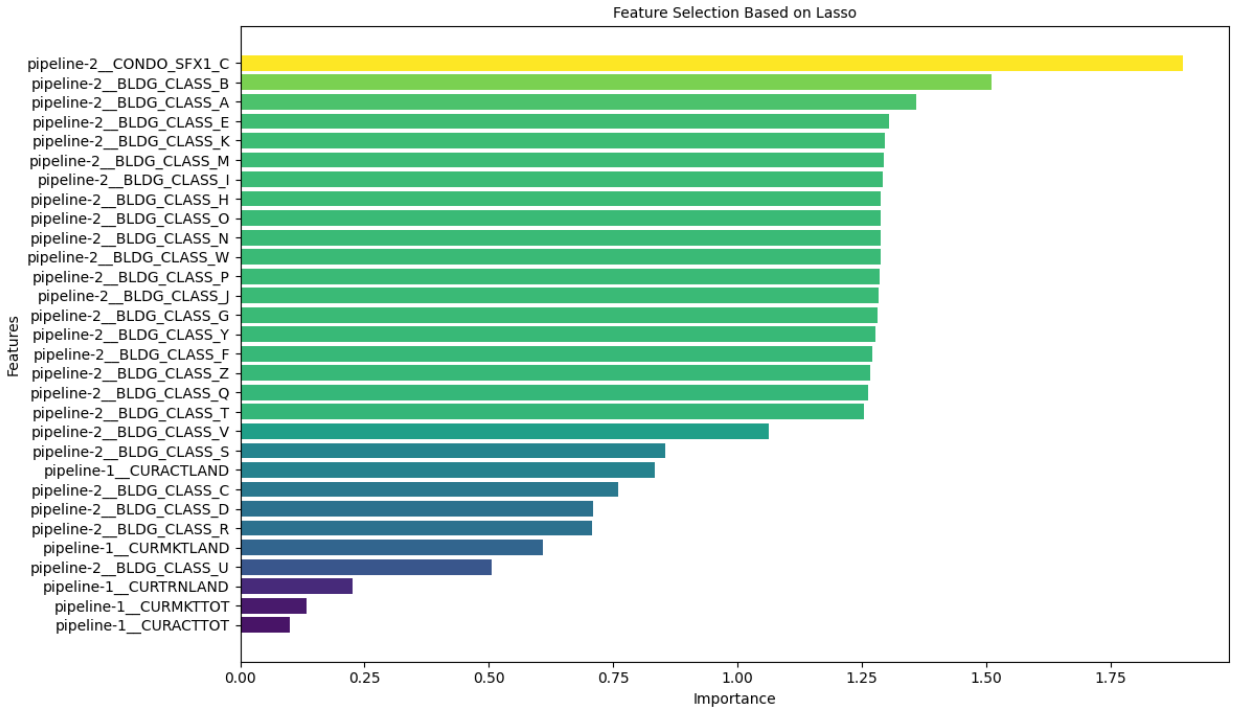
## Model Results

Our analysis mainly focused on two parts which include finding the features that are highly important in predicting the Tax class and building models that predict the tax class of a property. We have relied on four different machine learning models to achieve our objective.

### *Feature Selection Results*

We have used Lasso Regression and Random Forest Classifier to get the list of features that are important for the prediction of tax class. Cross validation using GridSearchCV has been used to fine tune the parameters of lasso which is alpha. We have used a five-fold cross validation that has given alpha = 0.00001 as the best parameter for the dataset. Below is the bar plot to depict the top 30 features as per the penalty outputs of lasso.

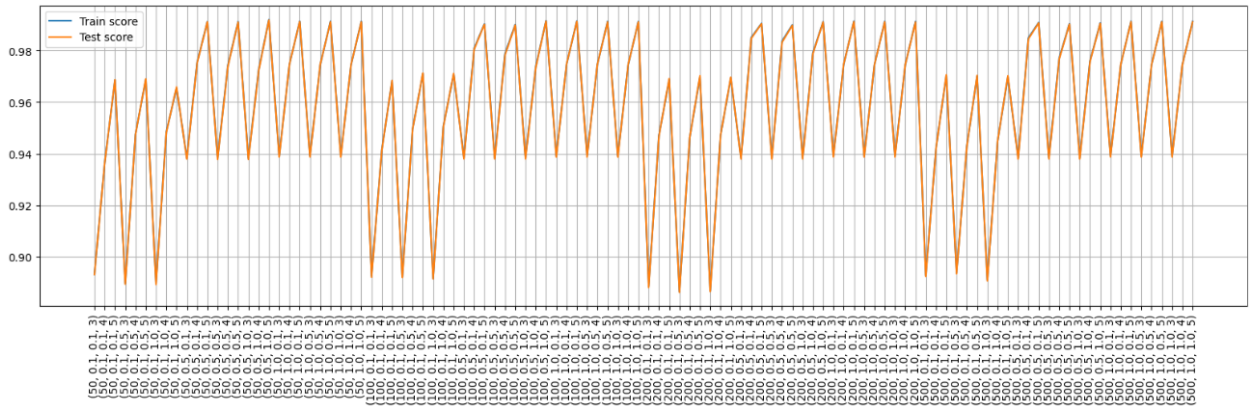
Figure 4: Top Features Based on Lasso Regression




As per the results and the plot, CONDO_SFX1_C is the most important feature that determines the target variable. It is interesting to note that all the dummy variables of BLDG_CLASS (Building Class) are among the top feature set along with other numerical columns such as CURMKTLAND, CURMKTOT, CURACTTOT.

Iterative method has been chosen to fine tune the hyper-parameters of Random Forest Classifier. Parameters such as n_estimators, max_features, max_samples, max_depth have been used to fine tune the model with values in range [50, 100, 200, 500], [0.1, 0.5, 1.], [0.1, 0.5, 1.], and [3, 4, 5] respectively. Below line plot shows the model complexity across various values of the parameters.

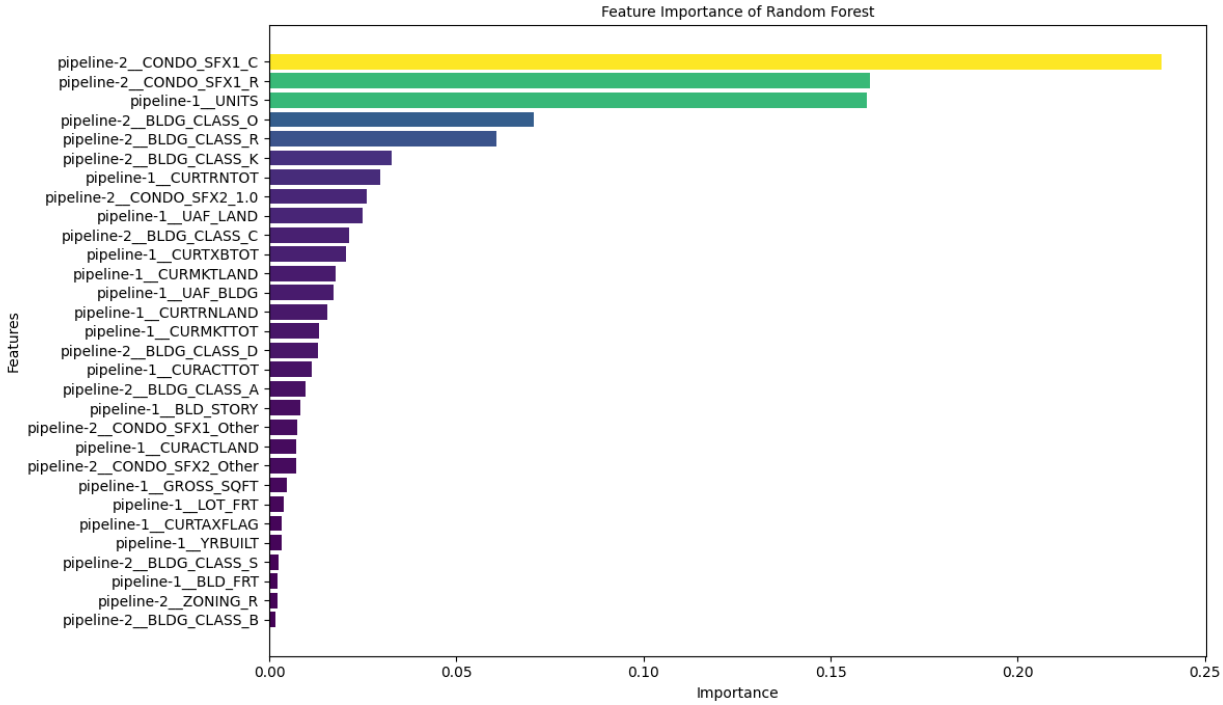
Figure 5: Model Complexity Chart of Random Forest Classifier




From the chart, we can say that the train and the test scores are almost similar, and the minimum score is near to 90% accuracy. Hence, the model performs well and generalizes the patterns in the dataset to unseen data. Based on the plot we can chose [50, 0.5, 0.1, 5] as the best set of parameters that give high accuracy. Further results about other scores and comparisons with other prediction models are done in further detail in the coming section.

After the model has been built with the best possible parameters for the current dataset, feature_importances_ to the values and have been mapped to corresponding columns to create a bar plot to display the top 30 features as per the model.

Figure 6: Top 30 features as per the Random Forest Classifier Model




As per the chart, CONDO_SFX1 is the most important feature followed by UNITS and other dummy variables of BLDG_CLASS, YRBUILT, UAF_BLDG. The top 3 features except the first vary in both the models where lasso has been dominated by BLDG_CLASS dummies, Random Forest has UNITS in it. It is also interesting to note that various other features are also being highlighted such as UNIT, YRBUILT, LOT_FRT, CURTAXFLAG. But the most important feature is CONDO_SFX1 in both the models.

***Prediction Models Comparison***

As mentioned in the previous section, Logistic Regression, Decision Tree, and Random Forest models have been used to predict the target and find the best possible model for the current dataset.

For the model evaluation, due to class imbalance, class 2 has almost 83% of the instances in the dataset. Most of the models are giving high accuracy scores due to the imbalance and because of the feature BLDG_CLASS which is highly correlated to the predictor. Hence, for the analysis of the model performance, we chose to focus on the precision score to reduce the FP (False Positives) thereby increasing the correct predictions.

Lasso Regression model has also been fine-tuned based on the max_iter parameter. 35 has been chosen as the best possible value that gives higher accuracy and good precision scores. Below are the charts of confusion matrix of train and test with classification report.

Figure 7: Confusion Matrix of Logistic Regression

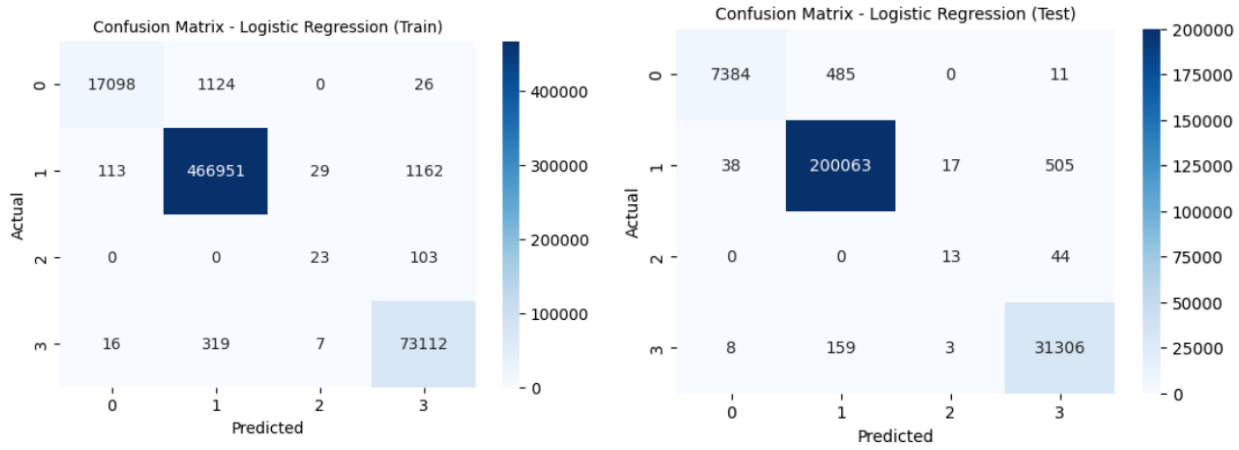


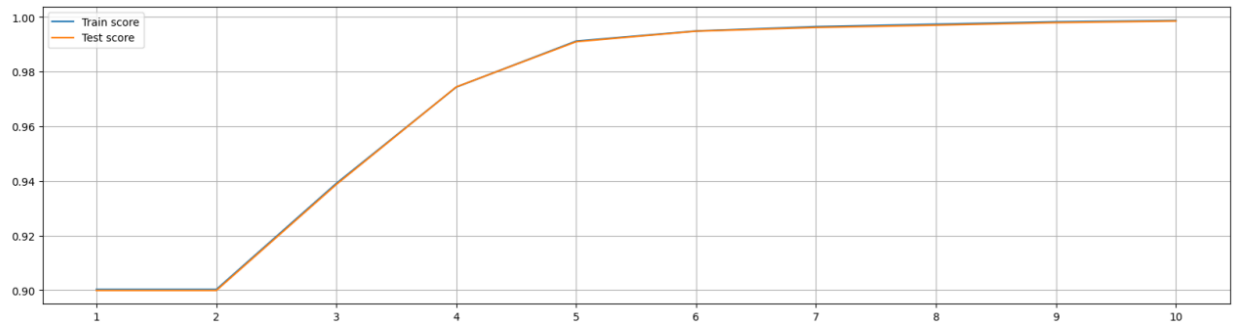

Figure 8: Classification Report of Logistic Regression

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1 | 0.99 | 0.94 | 0.96 | 7880 |
| 2 | 1.00 | 1.00 | 1.00 | 200623 |
| 3 | 0.39 | 0.23 | 0.29 | 57 |
| 4 | 0.98 | 0.99 | 0.99 | 31476 |
| accuracy | | | 0.99 | 240036 |
| macro avg | 0.84 | 0.79 | 0.81 | 240036 |
| weighted avg | 0.99 | 0.99 | 0.99 | 240036 |

From the results of Confusion Matrix, we can say that logistic regression model has been able to predict some instances of class 3 in both train and test datasets. This result can also be observed in the Classification Report which shows a great accuracy of nearly 0.00 with macro-precision score being almost 0.84. Due to high class imbalance in the dataset, macro averaging scores are being considered for precision.

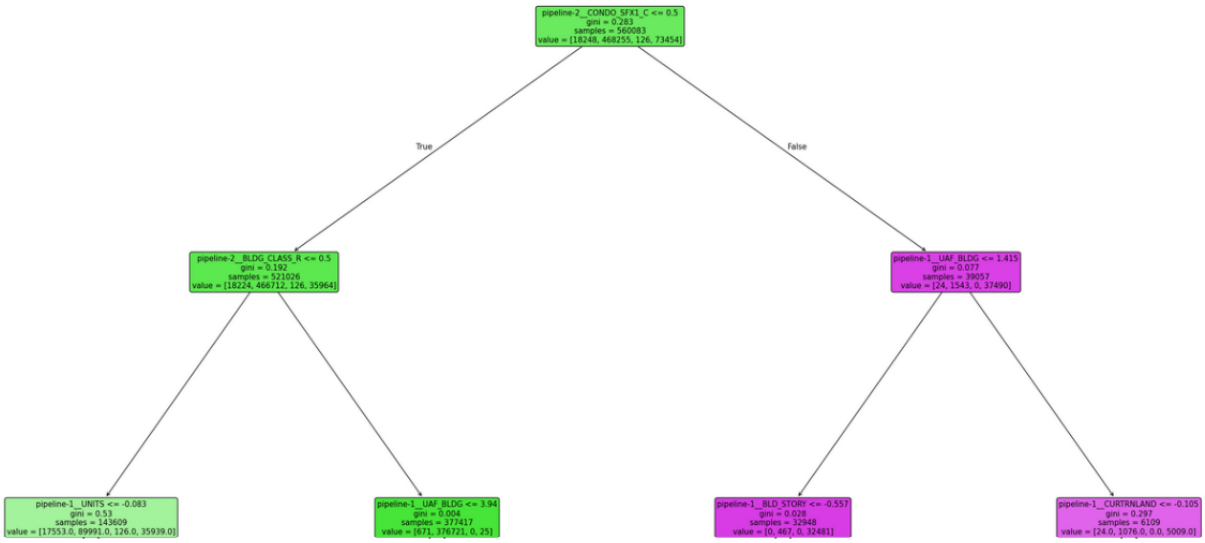
Fine tuning of Decision Tree parameter max_depth has been done iteratively. Values ranging from 1 to 10 have been iterated and a plot of corresponding train and test accuracy for each value respectively has been plotted to compare and get best possible value.

Figure 9: Model Complexity chart of Decision Tree




From the chart, we can see that there is no overfitting the model as the train and test scores are similar at various levels of max_depth. We chose max_depth as 4 to get the best accuracy that is not too high but enough to get good precision scores for each class. It is interesting to note that the accuracy starts from nearly 0.90 for the first root node itself. Which means that the root node alone can explain 90% of the data patterns of the target. Below is the tree plot that shows the way of dividing the nodes into sub-nodes to get a tree-like structure.

Figure 10: Tree Plot of Decision Tree




As per the tree plot, CONDO_SFX1_C is the root node that has the highest Information Gain, based on which division will take place. As mentioned above, it is the most important feature which is also depicted by Lasso and Random Forest feature importances. Below are the results of the decision tree model which includes the confusion matric and classification report.

Figure 11: Confusion Matrix of Decision Tree

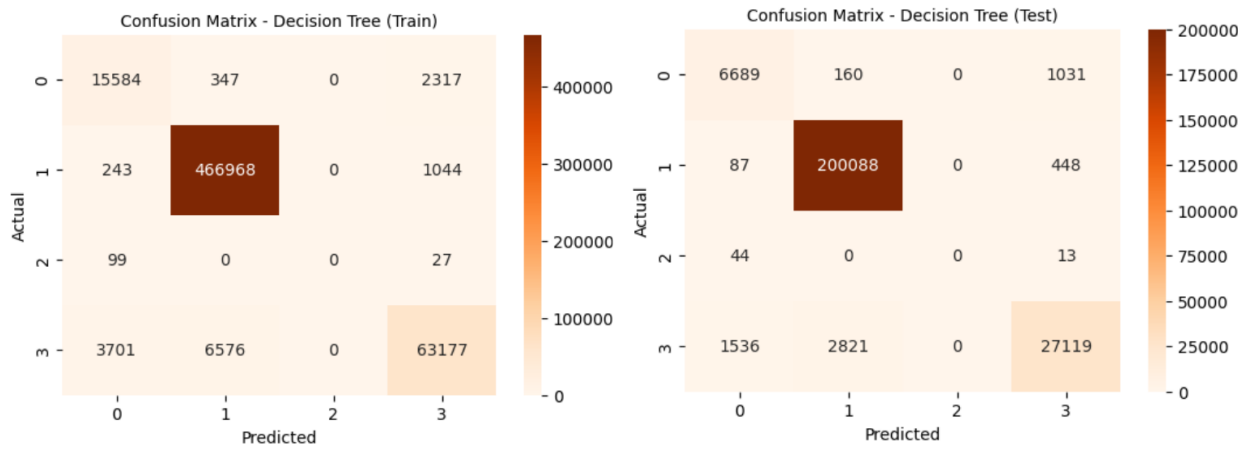


Figure 12: Classification Report of Decision Tree

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1 | 0.80 | 0.85 | 0.82 | 7880 |
| 2 | 0.99 | 1.00 | 0.99 | 200623 |
| 3 | 0.00 | 0.00 | 0.00 | 57 |
| 4 | 0.95 | 0.86 | 0.90 | 31476 |
| accuracy | | | 0.97 | 240036 |
| macro avg | 0.68 | 0.68 | 0.68 | 240036 |
| weighted avg | 0.97 | 0.97 | 0.97 | 240036 |

From the results, the model is not predicting the Class 3 instances in both the train and test datasets and hence likely unable to do so on the unseen data. The macro-precision score of the model is nearly 0.68 which is much less than the logistic regression model.

As explained in the Research Methods section, Random Forest is an ensemble machine learning model that works on multiple decision trees coming together. For our dataset, the decision tree model was unable to predict class 3 instances and thereby reduced macro-precision score. Hence, it might be possible to get similar results using the Random Forest model with improved accuracy.

Parameter tuning of the model has been explained in the previous subsection while explaining the feature selection results. Below are the model analysis results that include confusion matrix and classification report.

Figure 13: Confusion matrix of Random Forest

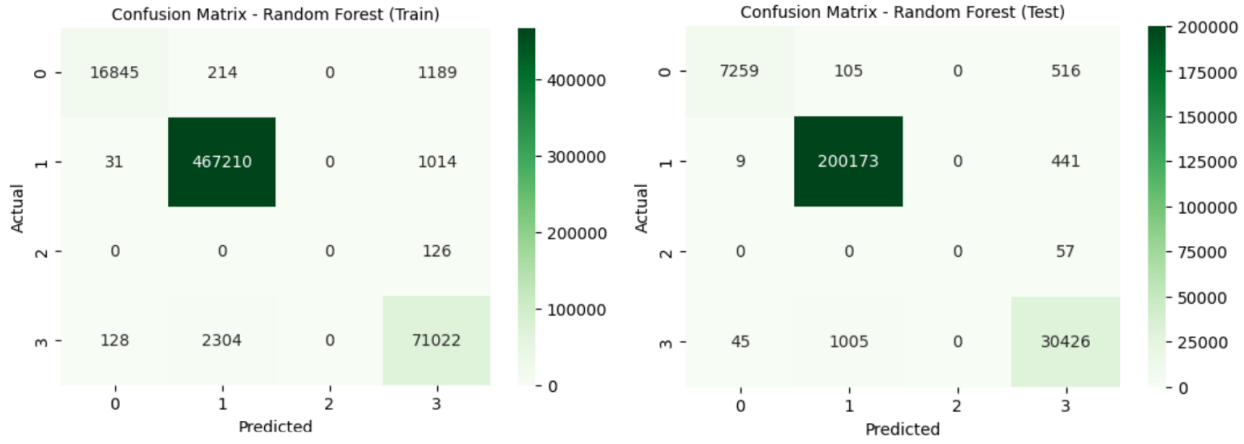



Figure 14: Classification Report of Random Forest

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1 | 0.99 | 0.92 | 0.96 | 7880 |
| 2 | 0.99 | 1.00 | 1.00 | 200623 |
| 3 | 0.00 | 0.00 | 0.00 | 57 |
| 4 | 0.97 | 0.97 | 0.97 | 31476 |
| accuracy | | | 0.99 | 240036 |
| macro avg | 0.74 | 0.72 | 0.73 | 240036 |
| weighted avg | 0.99 | 0.99 | 0.99 | 240036 |

As mentioned above, the random forest model gave a better accuracy of nearly 0.99 but was not able to predict class 3 instances. Though macro-precision scores have increased compared to the decision tree model, which is nearly 0.74, it is still not satisfactory.

## Discussion

This study aimed to uncover the connection that exists between property characteristics and tax classification with the help of predictive modeling tools. The results show that a particular set of factors, especially CONDO_SFX1 and BLDG_CLASS, have a strong influence on the property regulation of CURTAXCLASS. The same variables were showed as the most important

predictors according to the feature selection results... This result agrees with the proposal put forward earlier that tax classification largely depends on the structure of a property.

On the contrary, these findings confirmed that exemption-related variables like CURACTEXTOT, CURTRNEXTOT, and CURTXBEXTOT are ineffective in changing tax class distributions. Among those, none of the features were named in the top 30 many important variables, which indicates very small influence on the target variable. That is, while indeed, exemptions may impact the last tax bill, they do not play a major role in the process of the classifications themselves.

Logistic regression was found to be the most effective of the predictive models tested. It is interesting to mention that logistic regression stood as the lone model that was able to accurately classify class 3 instances out of all other classes, despite being the least represented in the sample. Moreover, the macro-precision score for logistic regression was the highest, around 0.84, which means that it was able to give balanced performance to all tax classes, despite the inherent imbalance in the classes.

Nevertheless, it is important to note that the present study has a few areas where caution is warranted. The biggest concern is the fact that the models are not very interpretable, particularly in situations where decision-making requires a clear understanding of the process. Moreover, the lack of balance in class representation can uplift overall accuracy scores by covering up the fact that the models perform terribly on minority classes. Noteworthy also is the fact that the data used was only from public sources, and that the reliability of the models depends largely on the quality and the completeness of the data. On the other hand, factors such as rules on exemptions, changes in the law, or the change in the economic landscape have not been considered as likely causes of

property tax classifications over time, but are possible factors, which might have a wider significance for the analysis.

## Conclusion

This project sets out to analyze the factors influencing property tax classification using machine learning techniques and to evaluate the predictive performance of various models. The results provide clear insights into the structural features that most significantly impact tax classification, with CONDO_SFX1 and BLDG_CLASS standing out as the strongest predictors of the current tax class. Conversely, exemption-related variables were shown to have minimal influence, indicating that they do not play a substantial role in how tax classes are assigned.

The application of machine learning models further demonstrated that logistic regression provides the most reliable performance among the tested algorithms. Its ability to correctly classify even underrepresented classes, combined with a strong macro-precision score, emphasizes its robustness and practical utility in real-world scenarios.

Despite these promising outcomes, the study also recognizes its limitations. The results are constrained by the interpretability of machine learning models, the imbalance in class distribution, and the quality of publicly available data. Furthermore, the analysis did not account for external policy or economic changes, which could influence tax classifications in ways not captured by the model.

Overall, this research contributes to a deeper understanding of the variables that influence property tax classification and highlights the potential of predictive modeling in urban policy analysis. Future work could expand on this foundation by incorporating more diverse data sources, addressing model interpretability, and exploring causal relationships to better support data-informed decision-making.

## References

Agrawal, S. (2023, June 5). *Feature Selection Using Lasso Regression*. Retrieved from Medium: https://medium.com/@agrawalsam1997/feature-selection-using-lasso-regression-10f49c973f08

Avenue Law Firm. (n.d.). *How are Property Taxes Determined in New York City*. Retrieved from Avenue Law Firm: https://www.avenuelawfirm.com/property-taxes-determined-new-york-city/#property-classifications-in-nyc

Baladram, S. (2024, August 30). *Decision Tree Classifier Explained: A Visual Guide with Code Examples for Beginners*. Retrieved from Medium: https://medium.com/data-science/decision-tree-classifier-explained-a-visual-guide-with-code-examples-for-beginners-7c863f06a71e

GeeksforGeeks. (2024, April 10). *Pruning Decision Trees*. Retrieved from GeeksforGeeks: https://www.geeksforgeeks.org/pruning-decision-trees/

Gloudemans, R., & Sanderson, P. (2021). The Potential of Artifical Intelligence in Property Assessment. *Journal of Property Tax Assessment & Administration, 18*(2), 87-106.

Grodecka-Messi, A., & Hull, I. (2022). *Measuring the Impact of Taxes and Public Services on Property Values: A Double Machine Learning Approach.* Sveriges Riksbank.

Lee, C. (2022). Training and interpreting machine learning models: application in property tax assessment. *Real Estate Management and Valuation, 30*(1), 13-22.

McCord, M. J., Davis, P. T., Bidanset, P. E., & Hermans, L. D. (2022). Prediction Accuracy for Property Tax Mass Appraisal: A Comparison Between Regularized Machine Learning

and the Eigenvector Spatial Filter Approach. *Journal of Property Tax Assessment & Administration, 19*(2), 20-46.

NTPTS. (2024, March 14). *How the CAMA System Impacts Your Property Taxes*. Retrieved from North Texas Property Tax Services: https://ntpts.com/computer-assisted-mass-appraisal-system/

NYC Open Data. (n.d.). *Property Valuation and Assessment Data Tax Classes 1,2,3,4*. Retrieved from NYC Open Data: https://data.cityofnewyork.us/City-Government/Property-Valuation-and-Assessment-Data-Tax-Classes/8y4t-faws/about_data

Radha, N., & Lakshmi, R. D. (2011). Machine Learning Approach for Taxation Analysis using Classification Techniques. *International Journal of Computer Applications, 12*(20).

scikit learn . (n.d.). *Lasso*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html

Scikit Learn. (n.d.). *Decision Tree Classifier*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

scikit learn developers. (n.d.). *LogisticRegression*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Scikit-learn Developers. (n.d.). *RandomForestClassifier*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

# Appendix

## Data Dictionary

| Column Name | Column Description | Term, Acronym, or Code Definitions |
|---|---|---|
| CURMKTLAND | Current Market Assessed Land Value | |
| CURMKTTOT | Current Market Assessed Total Value | |
| CURACTLAND | Current Actual Assessed Land Value | |
| CURACTTOT | Current Actual Assessed Total Value | |
| CURACTEXTOT | Current Actual Exemption Total Value | |
| CURTRNLAND | Current Transitional Assessed Land Value | |
| CURTRNTOT | Current Transitional Assessed Total Value | |
| CURTRNEXTOT | Current Transitional Exemption Total Value | |
| CURTXBTOT | Current Taxable Assessed Total Value | |
| CURTXBEXTOT | Current Taxable Exemption Total Value | |
| CURTAXCLASS | Property Tax Class | |
| BLDG_CLASS | Building Class | There is a direct correlation between the Building Class and the Tax Class |

| | | |
|---|---|---|
| ZONING | Zoning code from NYC Department of City Planning | |
| GEOSUPPORT_RC | Status of the address data verification from Geosupport | Blank = No Geosupport processing attempted<br>00 = Verified by Geosupport<br>All other values = Geosupport exception existed |
| LOT_FRT | Lot Frontage in feet | |
| LOT_DEP | Lot Depth in feet | |
| LOR_IRREG | Irregular shaped lot | 1 = Irregularly shaped lot |
| BLD_FRT | Building Frontage in feet | |
| BLD_DEP | Building Depth in feet | |
| BLD_EXT | Extension | E = Extension<br>G = Garage<br>EG = Extension and Garage |
| BLD_STORY | The number of stories/floors for the building | |
| LAND_AREA | Total Land Area | If not 0, Total Land Area |
| NUM_BLDG | The Number of Buildings on the property | |
| YRBUILT | The year the building was constructed | |
| UNITS | | |
| CONDO_SFX1 | | C = Commercial unit |

| | | |
|---|---|---|
| | | R= Residential<br>Blank = Entire condo is either all residential or all commercial |
| CONDO_SFX2 | Suffix 1 sequence number | |
| UAF_LAND | Land percent of common interest in the entire condo | |
| UAF_BLDG | Building percent of common interest in the condo | |
| GROSS_SQFT | Gross Square Footage of the building | |
| CURTAXFLAG | Current Taxable Flag | T' = Taxable, 'A' = Actual or blank |

**Zoning Lookup Table**

| **Zoning districts** | **Description** |
|---|---|
| C | Commercial |
| R | Residential |
| OZS | Other Zones |
| M | Manufacturing |
| BPC | Battery Park City |
| NZS | No Zones |
| P | Park |

**Business Class Lookup Table**

| **Building Class** | **Description** |
|---|---|
| A | One Family Dwellings |
| B | Two Family Dwellings |
| C | Walk Up Apartments |
| D | Elevator Apartments |
| E | Warehouses |
| F | Factories And Industrial Buildings |
| G | Garages |
| H | Hotels |
| I | Hospitals And Health Facilities |
| J | Theatres |
| K | Store Buildings |
| L | Lofts |
| M | Religious Facilities |
| N | Asylums And Homes |
| O | Office Buildings |
| P | Indoor Public Assembly & Cult. Facilities |
| Q | Outdoor Recreational Facilities |
| R | Condominiums |
| S | Primarily Res. - Mixed Use |
| T | Transportation Facilities |
| U | Utility Bureau Properties |

| | |
|---|---|
| V | Vacant Land |
| W | Educational Facilities |
| Y | Government/City Departments |
| Z | Misc. Building Classifications |

## Code snippets of EDA

```
[ ] df["UNITS"].describe().apply(lambda x: format(x, 'f'))
```

| | UNITS |
|---|---|
| count | 1048575.000000 |
| mean | 6.226605 |
| std | 45.270616 |
| min | 0.000000 |
| 25% | 1.000000 |
| 50% | 1.000000 |
| 75% | 1.000000 |
| max | 8812.000000 |

dtype: object

```
[ ] df['NEWDROP'].value_counts()
```

| NEWDROP | count |
|---|---|
| 0.0 | 1025619 |
| 1.0 | 8001 |

dtype: int64

```
[ ] # Droping the column as the percent of True Values is
    df.drop('NEWDROP', inplace=True, axis=1)
```

```
[ ] df['NOAV'].value_counts()
```

| NOAV | count |
|---|---|
| 0 | 769990 |
| 0 | 278171 |
| Y | 356 |

dtype: int64

```
[ ] # Droping the column as the percent of True V
    df.drop('NOAV', inplace=True, axis=1)
```

```
[ ] df['VALREF'].value_counts()
```

| VALREF | count |
|---|---|
| Y | 630 |

dtype: int64

```
[ ] # Dropping the column as the percent of True Val
    df.drop('VALREF', inplace=True, axis=1)
```