Tree Census Capstone: Final Report

Project Proposal

Problem

Keeping track of the trees growing in the City of New York is a huge responsibility. Since there are over half a million trees throughout the city, keeping tabs on each one can prove to be a challenge. There are over 600,000 trees in the city, and every year, some get sick or die due to neglect and a lack of proper attention. In order to minimize this number, we will use tree census data provided by NYC Parks & Rec to create a model that predicts (strength of correlation) the health of a tree based on its location, trunk diameter, tree type, root condition, trunk condition, and branch condition (to list a few). That way, volunteers and city officials can respond in time to care for the ailing tree. The model will go a step further by taking into consideration where deceased trees are located and provide suggestions for where to grow new trees in order to maintain an evergreen city. We need to identify where and which trees have poor health.

Client

The client is the New York City Parks and Recreation Center, who are responsible for keeping track of all the trees grown within city limits. They are providing tree census data gathered from 2015 as an aid for creating a machine learning model that predicts the health of trees grown throughout the city. That way, NYC Parks and Rec can locate any ailing trees sooner in order to provide care while nursing them back to health.

Data

The data comes from NYC Open Data. Specifically, we are looking at the 2015 Street Tree Census. Data is collected by volunteers and staff from the NYC Parks and Rec Center. The columns we are interested in include tree ID, diameter, status, health, type of tree, trunk and root conditions, location, stewardship, sidewalks, guards, and root / trunk / branch conditions.

Using this data, we can create a machine learning model that predicts the health of a tree based on where the tree is located. In addition, we can also determine where new trees need to be planted.

Methodology

Data Wrangling

To begin, we will download the excel file and import the data into Jupyter Notebook. Then, after browsing through the first couple row, we begin the process of elimination so that we only work with the columns needed for analysis. During this time, we will also look for rows with missing / NaN values.

Data Storytelling and Visualization

Once the data is clean and ready to use, we use exploratory data analysis to take a closer look at the data. Numerical and categorical data are handled separately. During this process, our goal is to understand as much as possible about the data and gather supporting evidence. We will create data visualizations (bar graphs, line charts, box and whisker plots, etc.), test our hypothesis, draw inferences based on what we see, account for any pattern abnormalities and skewness, and apply normalization methods (data transformation) to skewed data. This phase relies on good use of statistical knowledge and finding statistical significance from the data. Using our best judgement, we can infer if the variables are correlated or if one implies the causation of the other. From there, we can start making a plan to model our data.

Statistical Analysis

After examining the data, we will model the data. Since this is a regression problem, this project requires a machine learning model and multiple linear regression in order to understand the relationship between multiple variables. Because the variables are plentiful, only relevant ones will be selected in order to reduce dimensionality.

Machine Learning

We will measure goodness-of-fit and use error scores (MAE, RMSE) to measure accuracy between predicted and actual data points. If there is any discrepancy between values, then we will use feature engineering and a random forest model to alleviate the problem. (Add a more technical explanation here) Training data, testing data, performance evaluation, understanding specific metrics that are used to test the model for accuracy (precision, recall, f1 score)

Finally, after modeling the data, we can interpret the data and provide appropriate conclusions for the clients. This part requires visual and written presentations that the client can easily understand and create actions for.

Deliverables

The deliverables in this project include a Jupyter notebook, PPT presentation, full written report, and Tableau data story.

Data Wrangling Procedure

Removing NaN / null Values and Duplicates

First, I imported the data into Jupyter Notebook. Then, I looked at the shape and first five rows of data. Since the dataset is large (683,788 rows and 45 columns), I searched for and removed any duplicates. There were none. Next, I checked for any rows that contained missing, null, or NaN values. Eleven columns contained missing / null / NaN values, but before we tackled that, we reassessed the purpose of this dataset and selected columns to remove that were irrelevant to the project.

Column Selections and Removals

After removing these columns, I looked at the 'health' column in order to see why some 31,616 rows are missing data. It turns out that trees with a 'Stump' or 'Dead' trees ('status' column) do not have the 'health' column filled in. I removed any trees that did not have a 'health' status. After removing these rows, I looked again at the list of columns to see if any rows still contained missing data. Only three columns contained missing data, 'spc_common,' 'spc_latin,' and 'problems.' Since there is no way to figure out what kind of tree is growing, I removed those rows. As for 'problems,' there is no way to figure out what kind of problems the trees are or are not having, so I remove those rows as well.

Removing Outliers

Since we are only looking at 'Alive' trees for this project, I removed the 'status' column since all values were the same. The dataset contained no more missing / null / NaN values. Next, I took a look at any outliers in the columns, starting with 'tree_dbh' or Tree Diameter.

Looking at the 'tree_dbh' column (also the sole numerical column), I noticed that the minimum was 0 and maximum was 425. Both values are outliers. Also, the 75% percentile was 16, a significant difference from the maximum. Trees cannot have a

diameter of 0 since then there would not be any tree, nor can they have a diameter of 425 since it's too big. The average diameter of a tree is around 100 with an occasional tree diameter going up to 150. However, since these are trees in NYC, it's rare that they will get that big. I imposed a limit between 1 and 100 for tree diameter to filter out any rows that contained a diameter less than or greater than the range. The plot of the new distribution was skill skewed to the right, so I have to normalize the data before analyzing it.

Statistical Analysis

After cleaning the 2015 Street Tree Census - Tree Data dataset and creating story visualizations, the next step is to analyze the data using statistics. The majority of data is categorical, which means there is text in place of integers, so to start off, I used a label encoder – one-hot label encoder – to convert the binary columns into 0 and 1s based on 'No' and 'Yes' answer choices, respectively. Non-binary columns would not benefit from one-hot label encoder since they are nominal, meaning there is no intrinsic order to the responses. For example, in the Guards column, the responses are 'None', 'Harmful', 'Helpful', and 'Unsure.' There is no relationship or order between these variables.

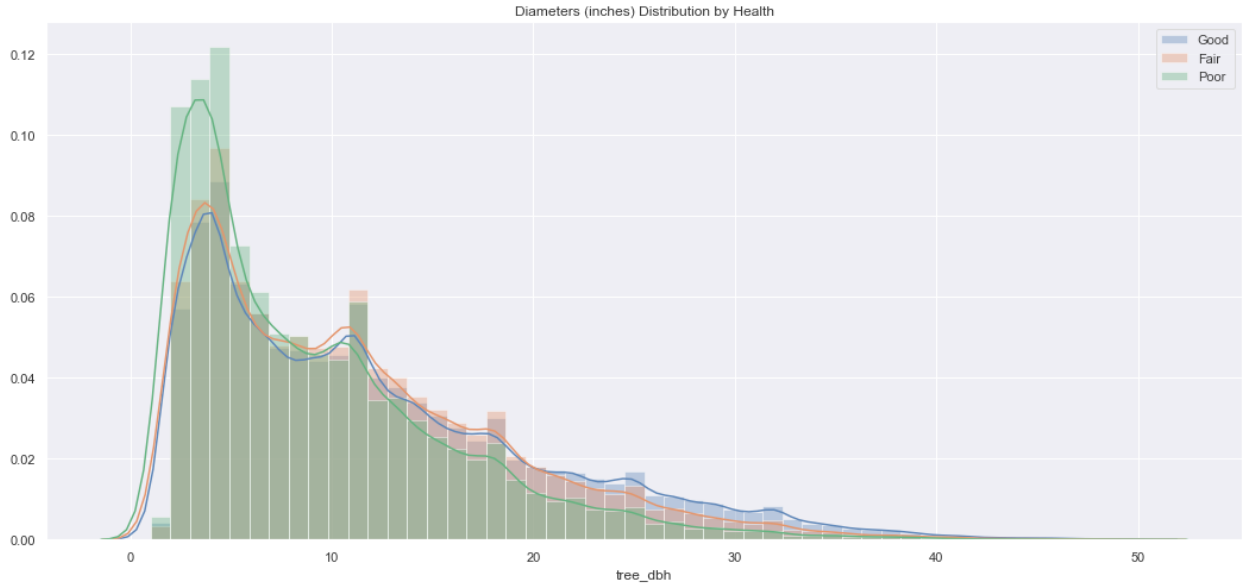
Since the majority of variables are categorical, I used the chi-squared test for association in order to determine if there is a statistically significant relationship between tree health and that variable. The null hypothesis states that there is no relationship between that column and tree health, or the health of a tree is not affected by that column. The p-value is 0.05.

Here are the results:

column	chi-square value	p-value	accept / reject
curb_loc	21.21	0.00074	Reject null
steward	82.64	1.01e-15	Reject null
guard	575.11	5.44e-121	Reject null
sidewalk	268.98	3.90e-59	Reject null
borough	1387.38	3.04e-294	Reject null
root_stone	602.17	1.74e-131	Reject null
root_grate	356.69	3.51e-78	Reject null

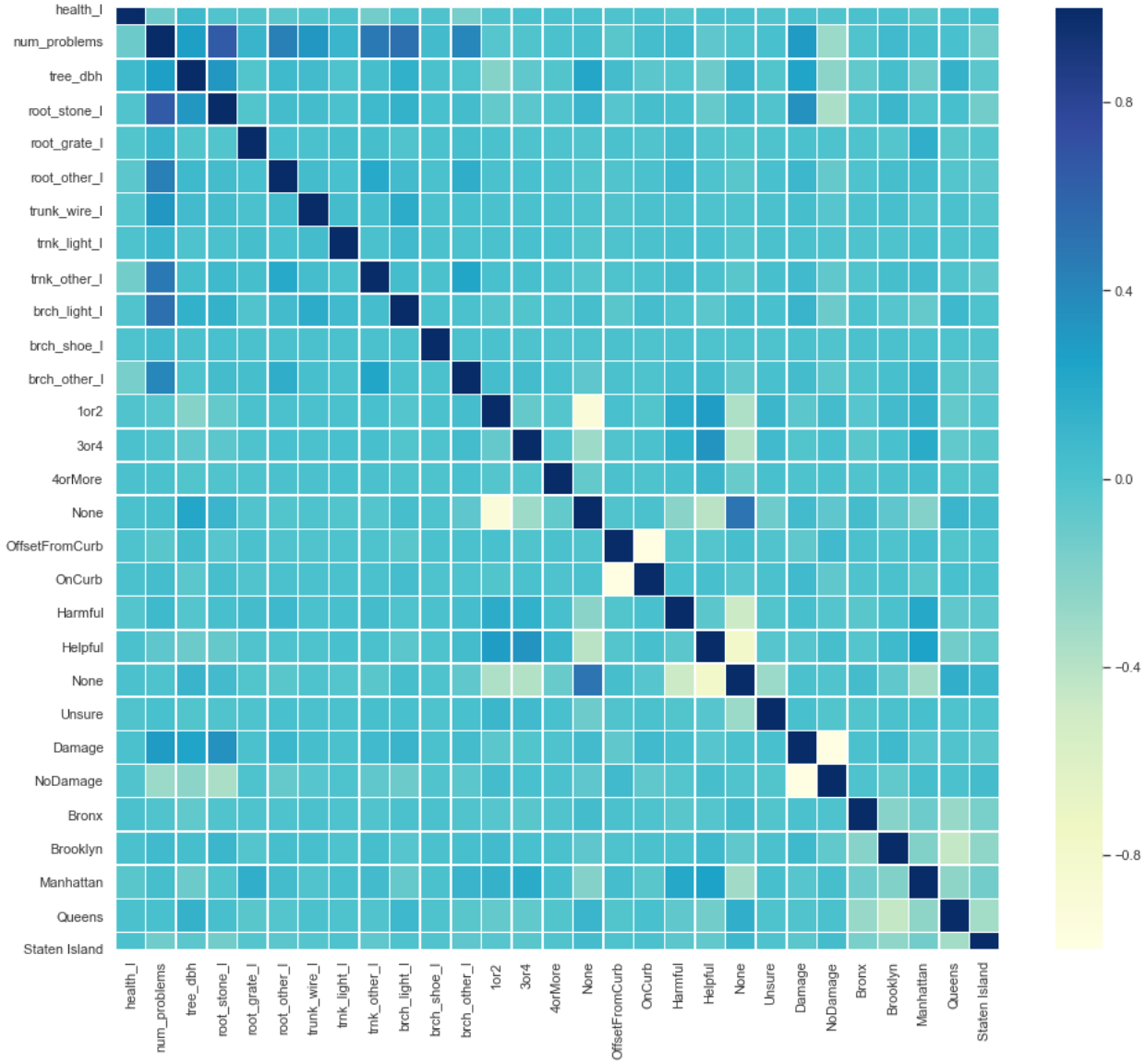
root_other	1928.30	0.0	Reject null
trunk_wire	511.11	1.03e-111	Reject null
trnk_light	42.65	5.47e-10	Reject null
trnk_other	11787.17	0.0	Reject null
brch_light	410.29	8.08e-90	Reject null
brch_shoe	35.44	2.02e-08	Reject null
brch_other	15111.46	0.0	Reject null

For the two numerical columns, tree_dbh and num_problems, we divided the data in each column based on the corresponding tree health. Since there are three categories, we used the analysis of variance, or ANOVA, test in order to compare the means of each group against each other. The ANOVA test is an extension of the t-test, which is used to determine statistical significance between the means of two variables.

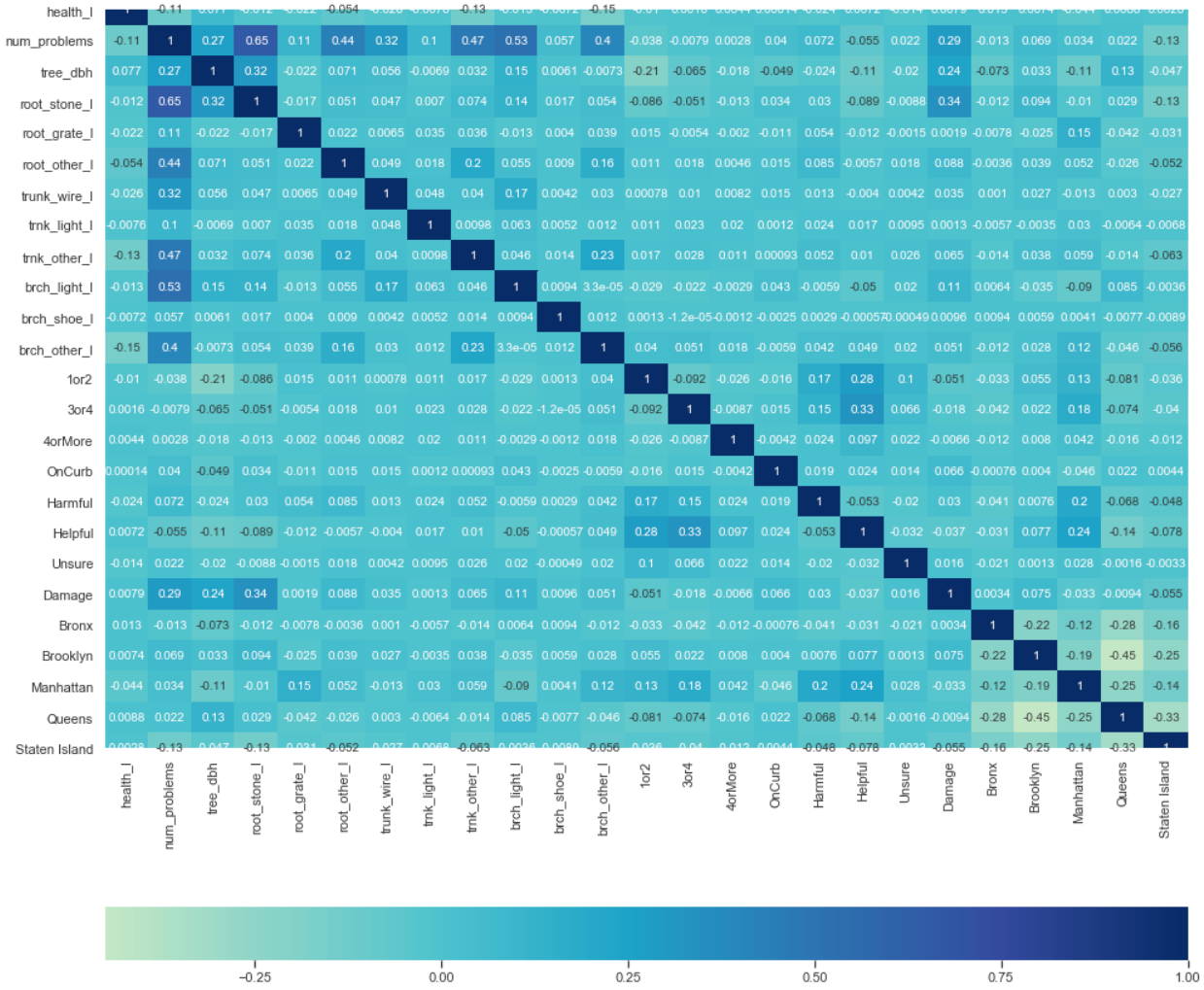


Finally, comparing the health column to each of the variable columns is not enough, as each variable column must also be compared to each other in order to establish that they are independent.

Here is the correlation table:



Looking into the correlations between independent variables, the columns OnCurb and OffsetFromCurb are directly inversely proportional to each other as are the Damage and NoDamage columns. Therefore, since only one column from each pair can be used, the OffsetFromCurb and NoDamage columns are removed. Another pair of columns, 1or2 and None, had a correlation factor of -0.92, which means that one of them must be removed. Since a fourth pair of columns, Helpful and None had a factor of -0.79, the None columns were removed. The final correlation table shows all variables as being independent of each other.



In summary, the statistical analysis shows that the relationship between the target variable, tree health, and its independent variables is statistically significant. This means that the data obtained is not an occurence of chance. For the categorical variables, I used the chi-squared test for association in order to establish that the variable is an influential factor in determining tree health. For the two numerical variables, I used the analysis of variance, or ANOVA, test, which separates the trees based on their health and compares the variable results to see if there is any difference between groups. For example, the test compares diameters of trees in each group to see if there is a difference between them. Trees in each group had noticeable differences based on their health group.

Since the majority of variables are categorical, I used label encoding and one-hot encoder to format the data into numerical columns so that machine learning techniques can be applied to it.

Machine Learning: In-depth Analysis

The purpose of machine learning for the tree census dataset is to locate a model that can accurately predict the health of a tree given its independent variables. During the visualization and statistical analysis phase, it became clear that the data is highly imbalanced. This means that one class has significantly higher representation than the others, and in this case, good trees are recorded far more frequently than fair and poor trees. With that in mind, we incorporated undersampling and oversampling methods as a way to give all classes a more equal representation since machine learning models tend to perform best when samples for each class are fairly similar.

Machine Learning

The algorithms we selected to fit the dataset to are logistic regression, KNN classifier, decision tree classifier, random forest classifier, gaussian naive bayes, and categorical naive bayes.

Under Sampler

For this dataset, we are using random under sampler, Tomek links, edited nearest neighbors, and near miss. Using Edited Nearest Neighbors, there are two models that fit the data very well, see below for results.

DecisionTree	Precision	Recall	f1-score
Fair	0.72	0.85	0.78
Good	0.66	1.00	0.98
Poor	0.91	0.49	0.64

RandomForest	Precision	Recall	f1-score
Fair	0.76	0.84	0.80
Good	0.96	1.00	0.98
Poor	0.94	0.49	0.65

Over Sampler

We used the over-sampling methods random over sampler, synthetic minority over-sampling technique (SMOTE), and adaptive synthetic (ADASYN).

Combination Oversampling and Under-sampling

There are three combination methods that we used: SMOTETomek, borderline SMOTE, and SVM Smote.