SPRINGBOARD DATA SCIENCE CAREER TRACK

CAPSTONE PROJECT #1

SILVIA MAIONE

Problem: Predicting the health of street trees in New York City

This project is based on a census conducted in 2015 by volunteers, staff and partners of the New York City Parks and Recreation department, who recorded data on almost 700k trees in the whole five boroughs' metropolitan area. The main goal is to provide a model to predict the health of a tree (good, fair or poor as categorized in the data set), knowing certain other characteristics collected in the survey. Other questions to which an answer is sought for regards for example the average diameter of trees, where there's the highest number of individuals, what the most common species are, etc. Also, more questions may arise as a result of exploratory data analysis.

Being able to identify what factors affect the trees' status allows proactive decision making. Beneficiary of this type of information are both the city's Park and Recreation department and the administrators. They can decide what actions need to be taken to guarantee the trees' good health and to prevent those that aren't in good conditions from deteriorating. Also, they can make an informed decision on where to plant new individuals and of what type.

Dataset description and data wrangling

The data used in this project can be found at the following page:

https://data.cityofnewyork.us/Environment/2015-Street-Tree-Census-Tree-Data/uvpi-gqnh

It consists of over 680k rows and 45 columns. The rows correspond to trees entries, the columns contain various forms of identification of a tree, including species' name; status of the tree (dead or alive, general health); information on size and other factors describing the surroundings (presence of guards, root or trunk problems, etc.). Most of the data is of object or text type, so it is converted to

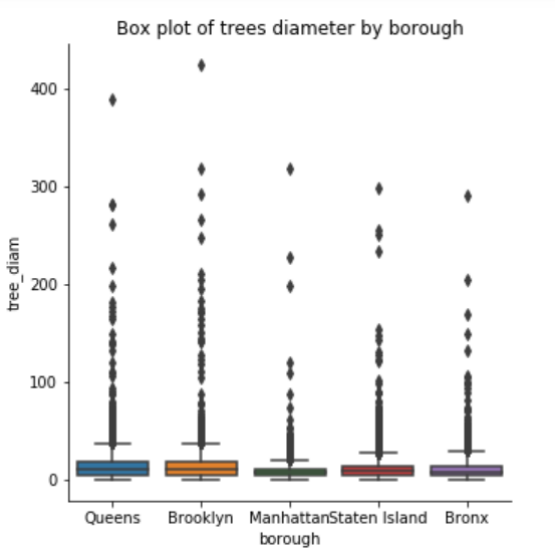
numerical to be potentially used in the model definition. Geographical information is also available (latitude, longitude). The target variable is chosen to be the general health of a tree, and it can have three values, i.e. good, fair or poor. It is available as one of the columns of the dataset. A snapshot of the data set is in the following picture (only a few rows and columns are displayed):

tree_id	block...	creat...	tree_...	stum...	curb_...	status	health	spc_l...	spc_c...	stew...	guards	side
180683	348711	08/27/2015	3	0	OnCurb	Alive	Fair	Acer rubr...	red maple	None	None	NoD
200540	315986	09/03/2015	21	0	OnCurb	Alive	Fair	Quercus ...	pin oak	None	None	Dam
204026	218365	09/05/2015	3	0	OnCurb	Alive	Good	Gleditsia ...	honeyloc...	1or2	None	Dam
204337	217969	09/05/2015	10	0	OnCurb	Alive	Good	Gleditsia ...	honeyloc...	None	None	Dam
189565	223043	08/30/2015	21	0	OnCurb	Alive	Good	Tilia amer...	American...	None	None	Dam
190422	106099	08/30/2015	11	0	OnCurb	Alive	Good	Gleditsia ...	honeyloc...	1or2	Helpful	NoD
190426	106099	08/30/2015	11	0	OnCurb	Alive	Good	Gleditsia ...	honeyloc...	1or2	Helpful	NoD
208649	103940	09/07/2015	9	0	OnCurb	Alive	Good	Tilia amer...	American...	None	None	NoD
209610	407443	09/08/2015	6	0	OnCurb	Alive	Good	Gleditsia ...	honeyloc...	None	None	NoD
192755	207508	08/31/2015	21	0	OffsetFro...	Alive	Fair	Platanus ...	London p...	None	None	NoD
203719	302371	09/05/2015	11	0	OnCurb	Alive	Good	Platanus ...	London p...	None	None	NoD
203726	302371	09/05/2015	8	0	OnCurb	Alive	Poor	Platanus ...	London p...	None	None	NoD
195202	415896	09/01/2015	13	0	OnCurb	Alive	Fair	Platanus ...	London p...	None	None	NoD
189465	219493	08/30/2015	22	0	OnCurb	Alive	Good	Platanus ...	London p...	3or4	Harmful	NoD

The data wrangling follows these steps:

- Remove some columns considered not relevant for the goal of this project, and rows not containing a value for the tree's health, since this is the target variable
- Identify and deal with missing data
- Check for duplicates
- Get initial statistical information
- Deal with potential outliers
- Convert text variables into numerical by label encoding

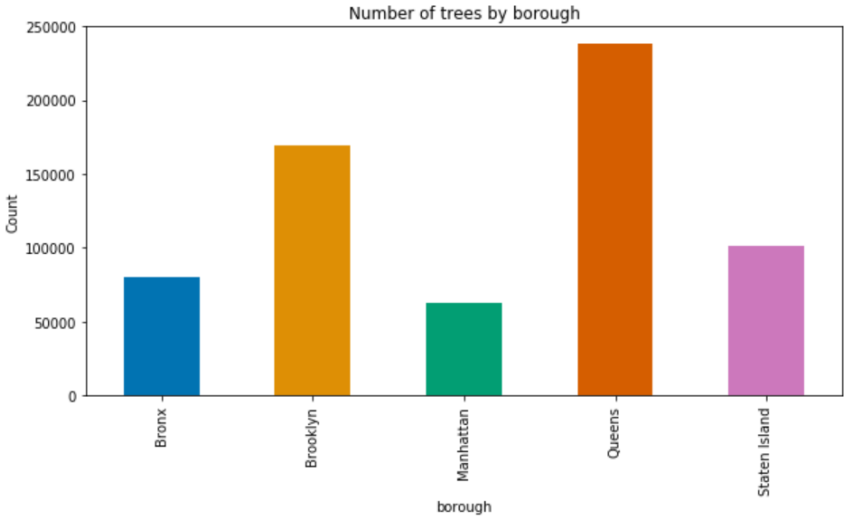
In particular, the rows with missing values are removed, since they are in limited number with respect to the total number entries. Also, the value of tree diameter in some cases is very high, as highlighted in the following box plot:

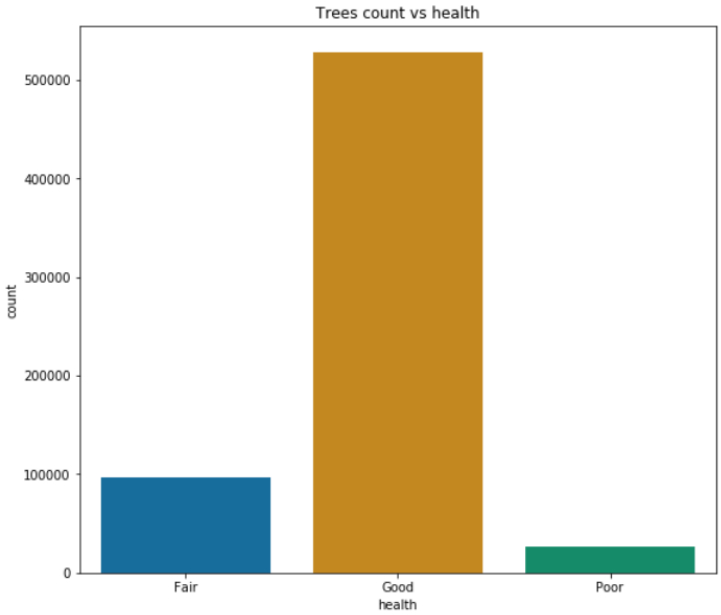


These values are considered entry errors (it's unlikely the diameter is this large), so the dataset is filtered by setting a threshold.

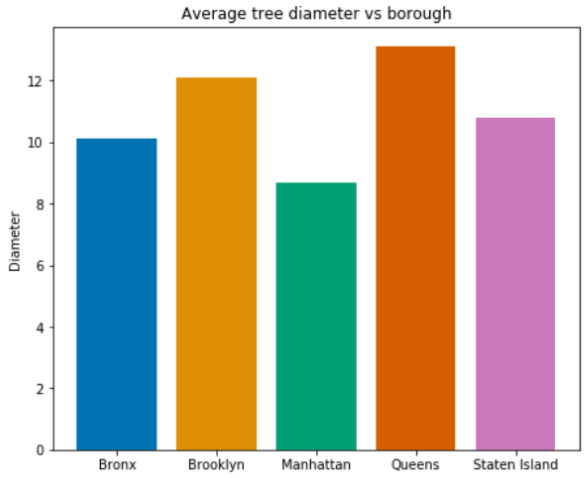
Exploratory data analysis

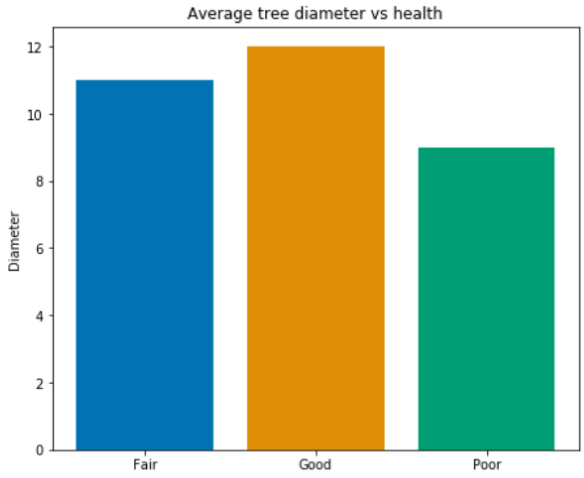
Most of EDA is done through visualization and statistical analysis. Since the health of the tree is the target variable, a certain number of plots include it. A couple of relevant findings are shown in the next two pictures, i.e. the highest number of trees isn't in Manhattan, and the dataset is skewed toward the "good" health:





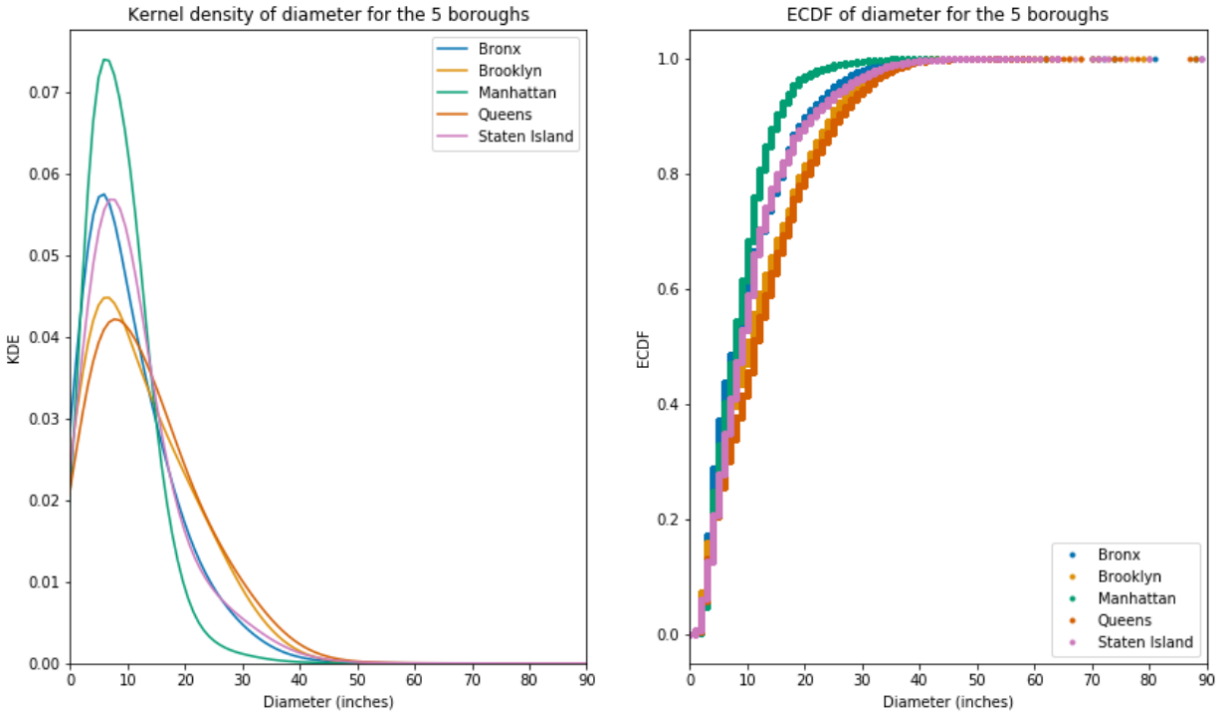
A variable of interest is the tree's diameter, since it is believed that it can be an indication of its health. A couple of plots seem to confirm that; moreover, where there are more trees, their diameter is bigger (see pictures below).



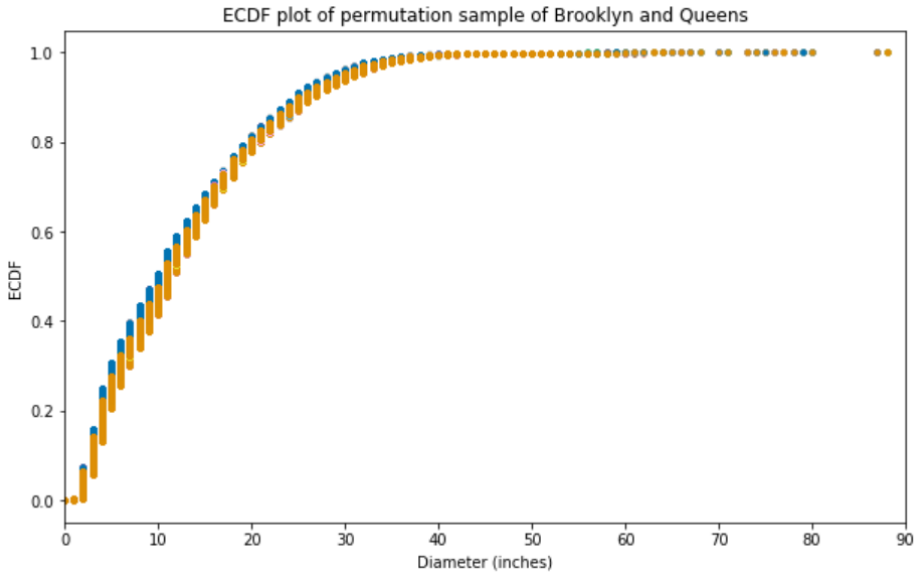


Also, there is some indication that trees having the same diameter but located in different areas are classified differently. Specifically, trees with larger diameter and located in bigger areas are healthier. This suggests that the model may need to be applied to the different boroughs separately or that the location should be included in the model. In fact, even some statistical analysis, specifically plotting the empirical cumulative density function for the five areas separately, shows that Manhattan has a different curve than the other boroughs, which are similar instead

taken two at a time. Also, the distributions look like a gamma distribution for all the boroughs, but with different parameters. So, certainly the diameter isn't normally distributed.



Permutation sampling is done to confirm that the diameter of trees located in two boroughs may come from a similar or same distribution. For example, the following figure shows the samples plot for the case of Brooklyn and Queens, (all the plots overlaps):



The other variables, which are categorical and include signs of stewardship, trunk and branches problems and similar, are believed to be associated to the tree's health. Any conclusion on that is drawn after statistical analysis, since visualization doesn't look to be enough for the purpose of understanding if they are correlated with the health or not. Instead, a chi-squared test of independence is run between health and the other categorical variables, one at a time, and all encoded. The results show a low p-value, below a confidence level of 0.05, for all of them. Also, the same test is run between these variables, two at a time, to see if there's any correlation. This is done for the five boroughs separately.

Machine learning model

With the selected features and target variable, the next step is to build a classifier. Again, the independent variable is the trees' health, which can belong to one out of three classes, i.e. good, fair or poor. Due to the findings from the data wrangling and statistical analyses, the model is trained and tested separately for the five boroughs. Another option would be to include the borough information in the dataset and build only one model.

For each borough, the dataset is split into training and test sets, the latter being used as a hold-out set for final validation.

Manly two types of classifiers are considered, i.e. Random Forest and KNN and their performances are compared.

Since the datasets are unbalanced, precision, recall and f1-score are the metrics used for evaluation and comparison. For the minority classes, recall is particularly important, since a high number of false negatives would mean classifying a lot of trees in poor health as in good shape. Vice versa, for the majority class, high precision is desirable, since a high number of false positive would mean some trees in fair or poor shape would be classified as in good health.

Cross validation is run on the training set. Both Random Forest and KNN give very high scores on training during cross-validation, but the scores on the hold-out set isn't good. The models are probably overfitting. For this reason, feature analysis is done using PCA, SelectKbest and RFE, confirming that the most important feature is the diameter. With these results in mind, a grid search is conducted to tune the

parameters on datasets containing both the tree diameter and only one of the other features are used. The scoring function used in the grid search is the f1-score; ideally precision would be optimized for the majority class and recall for the minority ones. For multi-class classification problems as this one only average scoring are available in the Python package used. In this case the "macro" option is chosen, so that the same importance is given to all classes.

Depending on what features are used, the recall of one of the minority class improves, but the recall of the other minority class gets worse. The original data contained also a "problem" column, which has been initially dropped since the columns with specific problem have been used. A grid search with only tree diameter and encoded problems column as features leads to results similar to the case where the specific problems columns are used. Overall, the results aren't good, mostly because the scores of the minority classes are very low. The following snapshot summarizes the results on the hold-out set by borough and for Random Forest classifier (KNN's are similar).

Manhattan								
"Fair" Class			"Good" class			"Poor" class		
Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
0.18	0.53	0.26	0.79	0.35	0.48	0.11	0.20	0.14
Brooklyn								
"Fair" Class			"Good" class			"Poor" class		
Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
0.16	0.26	0.20	0.84	0.4	0.54	0.05	0.51	0.1
Queens								
"Fair" Class			"Good" class			"Poor" class		
Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
0.15	0.52	0.23	0.84	0.44	0.58	0.09	0.15	0.11
Staten Island								
"Fair" Class			"Good" class			"Poor" class		
Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
0.16	0.21	0.18	0.83	0.73	0.78	0.06	0.15	0.09
Bronx								
"Fair" Class			"Good" class			"Poor" class		
Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
0.13	0.28	0.18	0.84	0.5	0.62	0.06	0.34	0.1

Conclusions

A multi-class classification problem is proposed in this project. The dataset is based on a tree survey held in New York city. The goal is to predict the health of a tree (fair, good or poor) based on other features. The dataset is unbalanced, with most trees in "good" health. From exploratory data analyses, it's clear that the most important feature is the tree diameter. Other features have also been considered to build a classifier and both Random Forest and K-Nearest Neighbor have been trained and tested. Overall, the two algorithms give similar results, i.e. predicting the majority class with 80% precision, while recall for the other two classes is about 50% and 20%, depending on the algorithm and the borough. These scores are on a hold-out set, put aside with the initial split of the data. The following options can be considered to improve the model:

- Investigate if the way the categorical variables are encoded has an effect, especially in the case of a Random Forest classifier
- Optimize the model with respect to different scores (precision, recall) depending on the class. Perform a custom grid search to tune the parameters, measuring the scores of interests per class, and not the average
- Combine the minority classes and make it a 2-class classification problem
- Reformulate the problem, for example as anomaly detection