Predicting Detroit Blight Ticket Compliance

STATS 415

— Zui Chen, Tiantian Ye, Xuewei Zhang

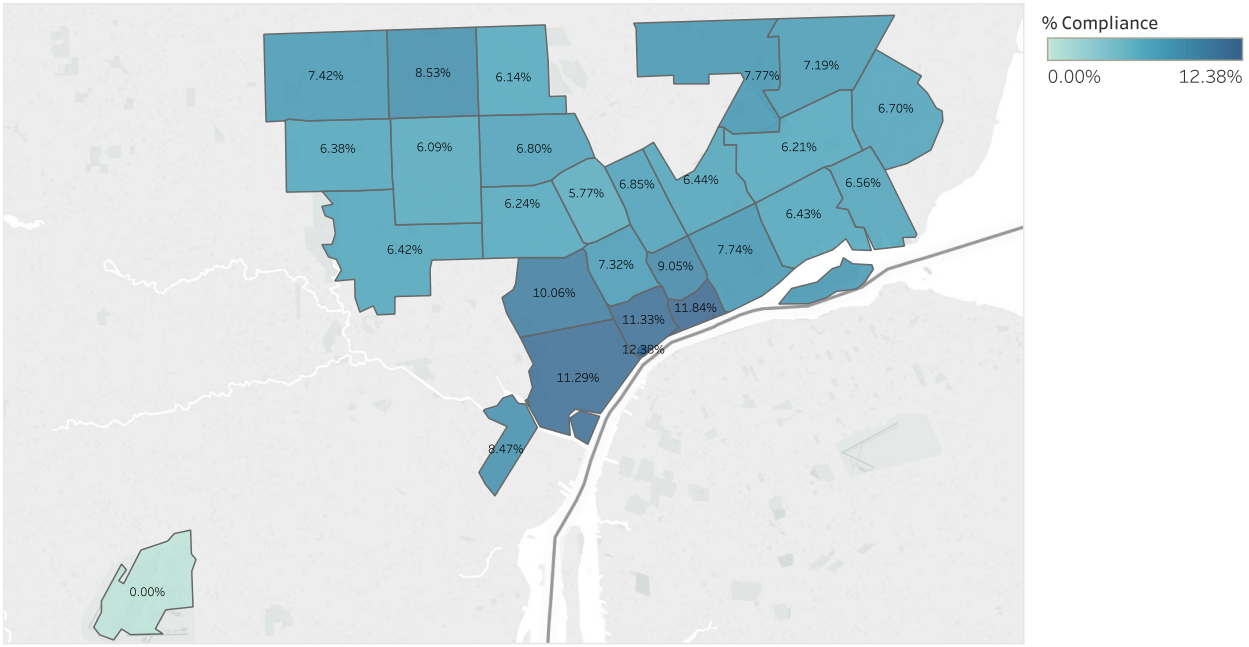
Introduction

Each year, Detroit imposes millions of dollars of fines on residents who fail to maintain their property, i.e. blight violations. However, many of these fines remain unpaid and it is costly and tedious to enforce these unpaid fines. The city could reduce a considerable amount of budget and improve efficiency by increasing the blight ticket compliance. We would like to find out the pattern of when and why a resident might fail to comply with a blight ticket in Detroit. We will look into the data and make a predictive model to anticipate whether a blight ticket will be paid on time.

Data Exploration

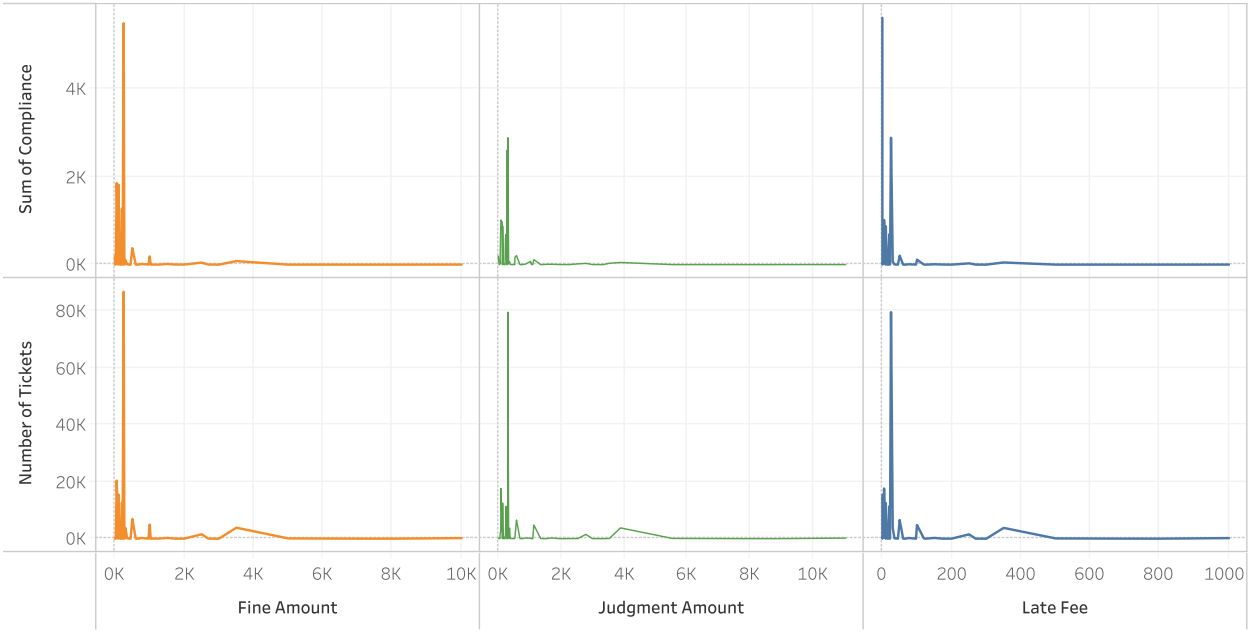
The data we use is provided by the Michigan Data Science Team (MDST) and the Michigan Student Symposium for Interdisciplinary Statistical Sciences (MSSISS) through the Detroit Open Data Portal (https://inclass.kaggle.com/c/detroit-blight-ticket-compliance/data). For single blight ticket, information about when, why, and to whom each ticket was issued is given. The response variable is compliance, which is “1” if the ticket was paid early, on time, or within one month of the hearing data, “0” if the ticket was paid after the hearing date or not at all, and “NA” if the violator was found not responsible. In the original dataset, there are 35 predicting variables, 28 of which are non-numerical variables. In order to reduce the dimensionality, we performed PCA on numerical variables and plotted correlation graphs between the predictors and response variables. We managed to reduce to 12 predictors, including zip, fine_amount, late_fee, disposition, judgement_amount, discount_amount, ticket_year, ticket_month, ticket_time, hearing_month, hearing_year, and violation_code, and recoded them for further analyses.

Violation Zip Codes

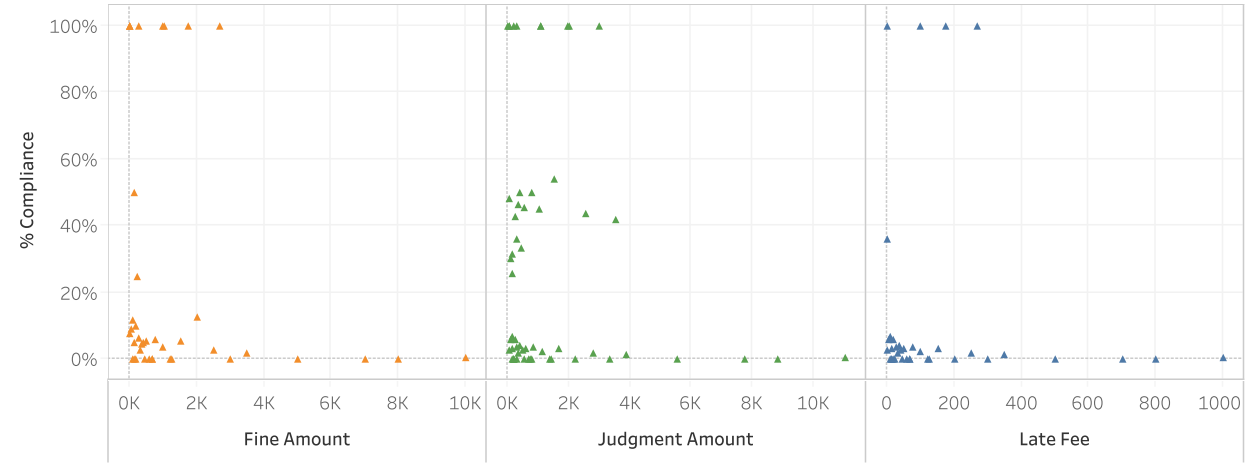


Map based on Longitude (generated) and Latitude (generated). Color shows SUM([Compliance])/COUNT([Compliance]). Details are shown for Zip. The view is filtered on Zip, which keeps 28 of 28 members.

Fees

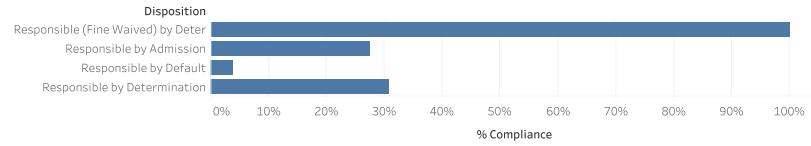


The trends of sum of Compliance and Compliance for Fine Amount, Judgment Amount and Late Fee. Details are shown for sum of Compliance and Compliance. The view is filtered on Judgment Amount and Fine Amount. The Judgment Amount filter ranges from 0 to 11030 and keeps Null values. The Fine Amount filter keeps all values.



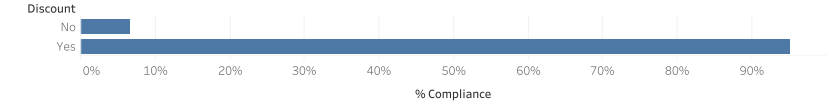
The plots of SUM([Compliance])/COUNT([Compliance]) for Fine Amount, Judgment Amount and Late Fee. The view is filtered on Judgment Amount and Fine Amount. The Judgment Amount filter ranges from 0 to 11030 and keeps Null values. The Fine Amount filter keeps all values.

Disposition



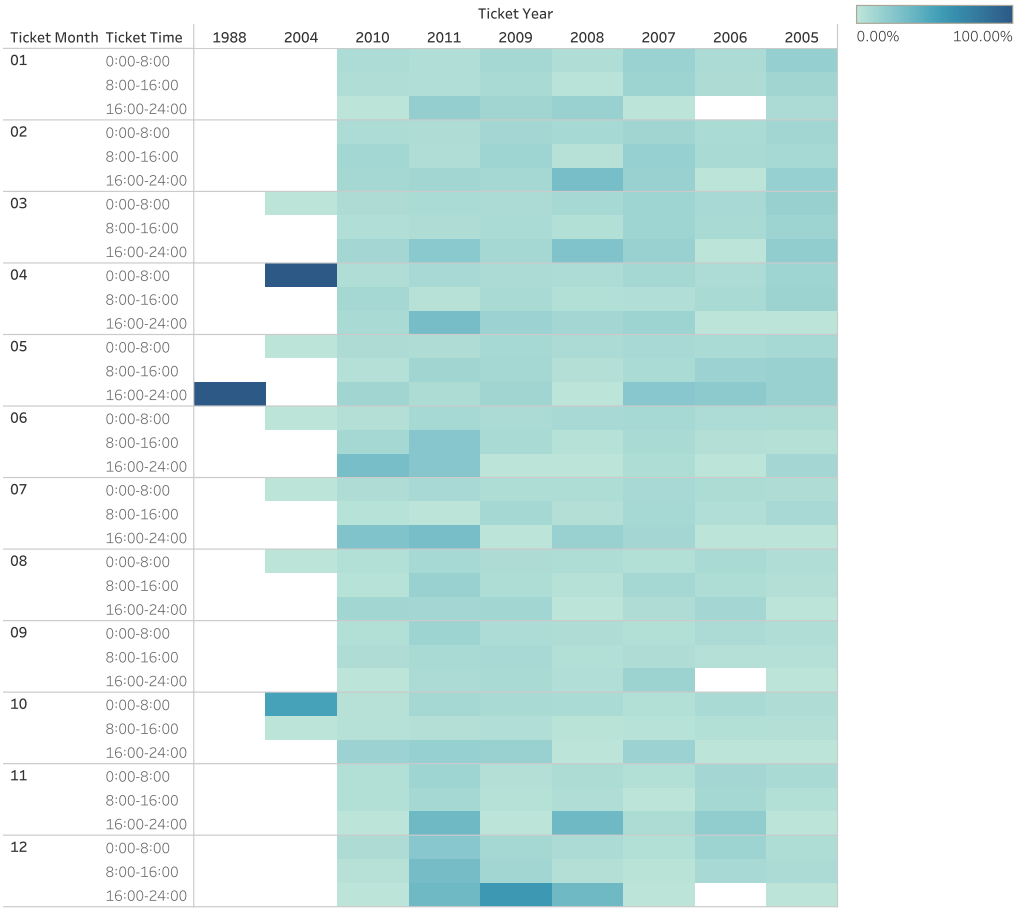
SUM([Compliance])/COUNT([Compliance]) for each Disposition.

Discount

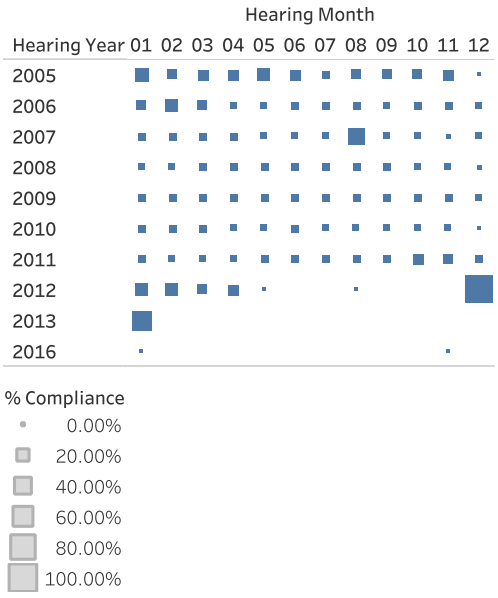


SUM([Compliance])/COUNT([Compliance]) for each Discount.

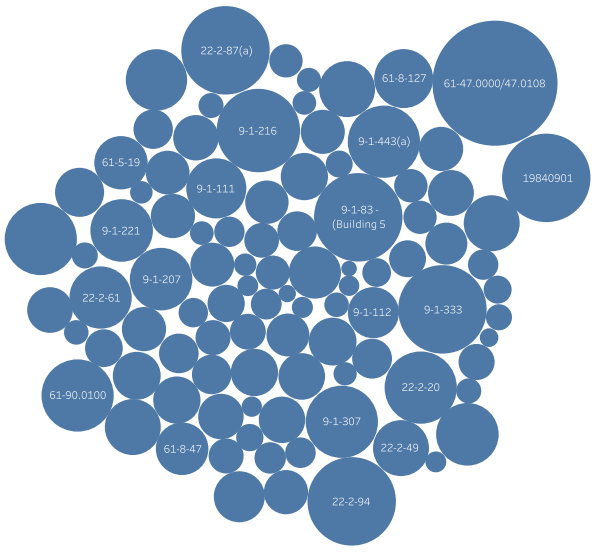
Ticket Time



Hearing Time



Violation Code



Size shows % Comliance. The marks are labeled by Violation Code.

The above graphs display the relationship between the percentage of compliance (percentage of “True” or “1” over total number of tickets in each level of each predictor) and the predictors. The violation zip code map shows a slight tendency of complying the tickets in southern Detroit. Among the four dispositions, “Responsible by Deter” has the full compliance rate. Besides, violators have a landslide in compliance when given a discount. Moreover, blight tickets for certain types of violation are more likely to be complied than the others.

Classification

We used LDA, QDA, Logistic Regression, CART, Random Forest, Bagging, Boosting and Generalized Additive model. When conducting LDA and QDA method, we focus on the quantitative variables and eliminate qualitative variables. In other methods, we used all the 12 predictors. We divide the dataset into train and test and fit models using different methods.

Parametric Method

We used parametric methods including LDA, QDA, Logistic Regression and Generalized Additive Model to find the relationship between predictors and response and use the fitted model to predict the test data set to see the performance of each model.

While the parametric formula for LDA and GAM are extremely complicated and hard to interpret, LDA and LR result in presentations of the relationship in more interpretable ways. Below are the coefficients of the LDA and LR models.

	variables	coefficients
1	judgment_amount	-0.402906819
2	fine_amount	0.409951681
3	balance_due	-0.002376124
4	late_fee	0.353562362

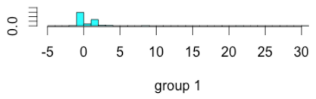
	variables	estimate	p_value
1	judgment_amount	0.2345	<2e-16 ***
2	fine_amount	-0.03116	<2e-16 ***
3	balance_due	0.03159	<2e-16 ***
4	late_fee	-0.0001741	<2e-16 ***
5	judgment_amount	0.02844	<2e-16 ***
6	fine_amount	0.7655	<2e-16 ***

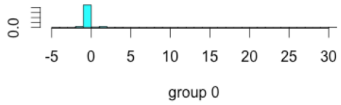
The resulting formulas are listed as following:

$$L(X) = -0.402906819 \times ja + 0.409951681 \times fa - 0.002376124 \times bd + 0.353562362 \times lf$$

$$P(X) = \frac{e^{(2.35e-01)-(3.12e-02)*ja+(3.16e-02)*fa-(1.74e-04)*bd+2.84*lf+(7.66e-01)*da}}{1 + e^{(2.35e-01)-(3.12e-02)*ja+(3.16e-02)*fa-(1.74e-04)*bd+2.84*lf+(7.66e-01)*da}}$$

In the LDA method, when L(X) is large, the classifier predicts 1 and otherwise predict 0. The histogram below shows the prediction of LDA. And on contrast, in the LR model, P(X) represents the probability of the response being 0 given the observation X and hence it predicts 0 when P(X) is greater than 0.5.





In the generalized additive model, we can see the underlying relationship between predictors and response. However, in our setting, there are 12 predictors and one of them is a qualitative variable having over 200 factor levels making the model equation extremely complicate as it concludes 12 functions each corresponding to one of the 12 variables but still it is possible for us to see the relationship between predictors and response. The table below summarized the p value of different variables of generalized additive models, note that disposition, late fee and discount amount have higher significant level.

	variable	p_value
1	violation_code	1
2	disposition	< 2.2e-16
3	fine_amount	0.6048514
4	late_fee	0.0037616
5	discount_amount	0.0001643
6	ticket_year	0.9969035
7	ticket_month	0.9997695
8	ticket_time	0.9159936
9	hearing_month	0.8962455
10	zip	1

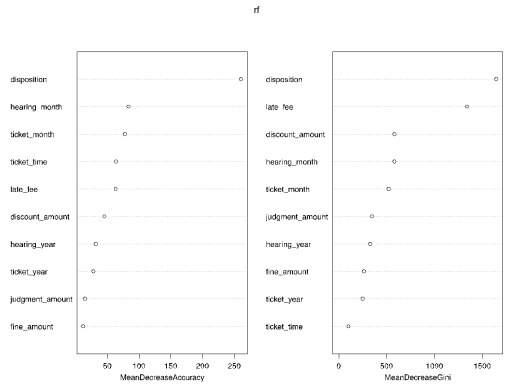
Nonparametric method

Despite the relatively good performance of ensemble methods in classification problems, the most important disadvantage of decision tree and tree-ensemble methods is that we don't know about the actual relationship between these predictor variables and the response when using nonparametric methods.

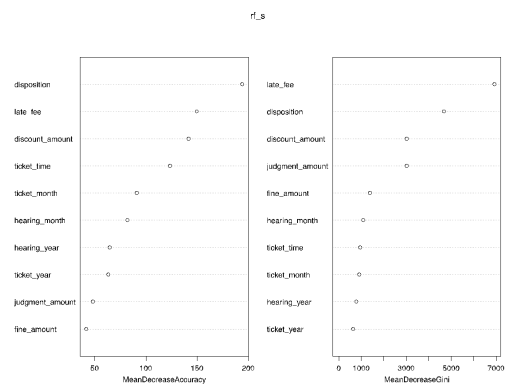
When applying KNN, which is a high nonlinear model, and dimension of this data is very high. If we put all the data, there are too many ties in KNN. Therefore, we extract a fraction of this data and run cross validation to choose k which is 4 in this case and use the fit to predict response in the test set and we will analyze its performance later.

When using tree based methods, including CART (classification tree), random forest, bagging and boosting, although we can't have the exact visualization of the relationship between predictors and response, however, we can still see the relative importance of different predictors in the datasets.

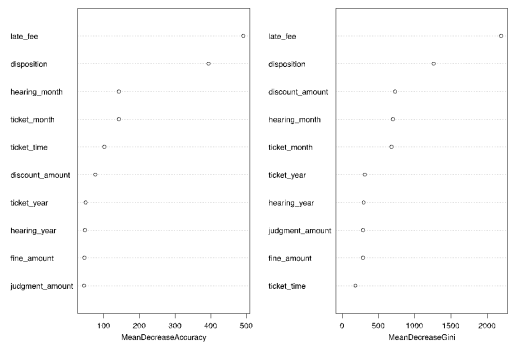
Plot 3 (random forest VarImpPlot)



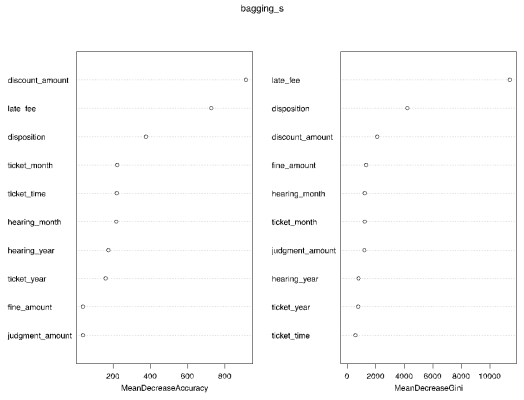
Plot 4 (random forest using smoteVarImpPlot)



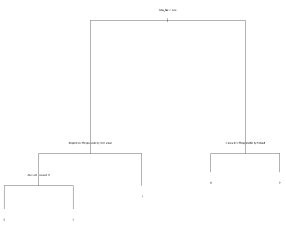
Plot 5 (Bagging VarImpPlot)



Plot 6 (Bagging using smote VarImpPlot)



Plot 7 (classification tree)



Plot 8 (classification tree using smote)

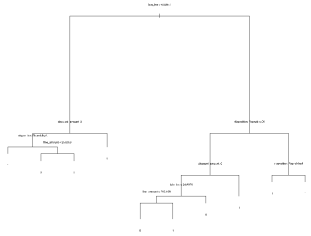


Table 14 (Boosting summary)

	var	rel.inf
late_fee	late_fee	50.21
discount_amount	discount_amount	21.85
disposition	disposition	21.73
violation_code	violation_code	3.40
judgment_amount	judgment_amount	0.90
hearing_month	hearing_month	0.67
zip	zip	0.51
fine_amount	fine_amount	0.38
hearing_year	hearing_year	0.24
ticket_month	ticket_month	0.10
ticket_year	ticket_year	0.00
ticket_time	ticket_time	0.00

Table 15 (Boosting using smote summary)

	var	rel.inf
late_fee	late_fee	44.42
disposition	disposition	23.37
discount_amount	discount_amount	20.14
ticket_time	ticket_time	5.36
violation_code	violation_code	2.75
hearing_month	hearing_month	1.46
hearing_year	hearing_year	1.29
ticket_month	ticket_month	0.42
fine_amount	fine_amount	0.35
zip	zip	0.20
ticket_year	ticket_year	0.12
judgment_amount	judgment_amount	0.12

From the plots above, we can see that when using decision tree, the outcome is highly variable depending on the input of the data. This is referred to the instability of the decision tree and tree-ensemble methods could reduce this instability significantly by combining different decision trees. We use the VarImpPlot command in the “random forest” package and the summary function in the “gbm” function to analyze the levels of importance of predictors in the ensemble methods. We see that different ensemble algorithms weight the same predictors with different weights and the different training data sets also leads to different interpretations of the relative importance of the same data. But overall, late fees, disposition and discount amount are the three predictors with the largest importance weights among all the other predictors. These 3 variables turn out to correspond to the variables that have higher significant levels in the generalized additive model in the previous section.

Results

As the data is highly unbalanced with more than 90% of the data belong to the class (Y=0), the performance of traditional methods including tree methods are influenced significantly by the unbalanced nature of the data. This is due to the fact that traditional classification methods when dealing with unbalanced data sets tend to exceed in classifying pattern as belonging to the majority class because of the peculiar characteristics of these classifiers and their algorithms. (Igelnik n.d.)

As tackling imbalance has received more attention in recent years, more and more methods have been developed to address the problem. One common solution to the problem of imbalance is to use over-sampling and under sampling to adjust the class distribution of a data set making it more balanced. “Synthetic Minority Over-Sampling Technique” (SMOTE) is one of the most popular algorithm in addressing the problem using oversampling and under sampling. We used the “DMwR” package in R to create a more balanced data set and use it to perform classifications.

Table1 (distribution of smote data)

	counts
0	42815
1	25689

Table2 (distribution of the original train data)

	counts
0	111233
1	8563

As presented in the tables above, the smote data set is much more balanced than the original train data set. We use the smote data to fit classifications using different methods and record the results in confusion matrixes which record the amounts of true positive (upper left entry), true negative (lower left entry), false positive (upper right entry) and false negative (lower left entry) of each classification methods. The larger the number on the diagonal entries of the confusion matrixes (which represents the amount of correct predictions), the better performance of the classification method. Below is an example a confusion matrix of applying random forest algorithm.

Table 5 (pruned tree using smote)

	0	1
0	36853	2106
1	136	762

Table 6 (random forest)

	0	1
0	35632	1607
1	1357	1261

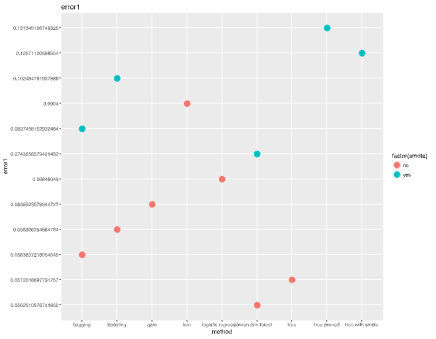
As we can see in the tables, although all the algorithms predict well when the true class of the response in the test data sets are 0, they all fail to predict accurately when the true class is 0. We then record the overall testing error as “error1” which is equivalent to 1-accuracy. The probability that the model predicts the response Y of an observation X to be 0 when the true response is the minority class “Y=1”, i.e. “type I error rate” or “false positive rate” as “error2”. The probability that the model predicts the response Y of an observation X to be 1 when the true response is the majority class “Y=0”, i.e. “type II error” or “false negative error rate” as “error 3”. We record these errors of all the models used and compare them in the table and plots below. In the two plots, testing errors using smote are denoted by green points whereas those using original train data are represented by red points.

Table 13 (Testing Error of different methods using original train set and smote data set)

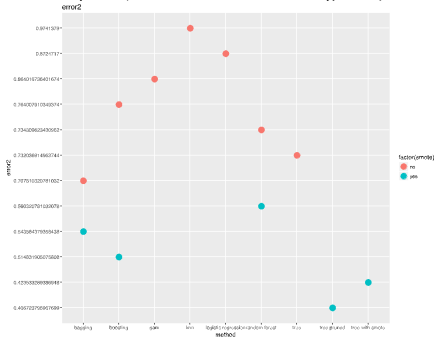
	method	error1	error2	error3	smote
1	tree	0.0572048697734757	0.732036914963744	0.00183678155054976	no
2	tree with smote	0.12571100688554	0.423533289386948	0.108556721899309	yes
3	tree pruned	0.131349166749825	0.406723796967699	0.117683122646192	yes
4	random forest	0.0562510976741852	0.734309623430962	0.00349084935444955	no
5	random forest	0.0743658579421432	0.560320781032078	0.0364402910926717	yes
6	bagging	0.0583837218054545	0.707810320781032	0.00767005836475389	no
7	bagging	0.0827458162932484	0.543584379358438	0.047243880572686	yes
8	boosting	0.058926254864784	0.764007910349374	0.00111890957176279	no
9	boosting	0.102484781957888	0.514831905075808	0.0705928020850663	yes
10	gam	0.0636525578944727	0.864016736401674	0.00149715793747462	no
NA	knn	0.0904	0.9741379	0	no
12	logistic regression	0.06646049	0.8724717	0.002762816	no

** error1 = total testing error rate; error2 = type I error rate; error3 = type II error rate*

plot 1 (error1 is larger when using smote)



plot 2 (error 2 is smaller when using smote)



Based on the output, it is clear that although all methods produce relatively small testing error, their performance is bad when the true responses belong to the minority class, i.e. Y=1.

We find that the overall error rates decrease when using smote as the under sampling of the original data results in loss of information of the data. However, when fitting the classification methods with the smote data sets, error2 are reduced significantly. Thus, all the classification methods perform better in predicting when the true response belong to the minority class which is Y=1 in this scenario.

We can conclude that the smote algorithm, which under samples the majority class observations as well as over samples the minority class enhance the predicting power of different classification algorithms when predicting the response of an observation which has the true response belonging to the minority class. However, the tradeoff is that it reduces its predicting power when the true response lies in the majority class. Thus, if the main interest lies in predicting the majority class, which is the probability that a ticket is non-compliant in the setting of the Detroit ticket data set, the original training data set would be a better choice. If, on contrast, the government is more interested in the probability that a ticket is compliant,

using the smote data set gives a more reliable result in predicting.

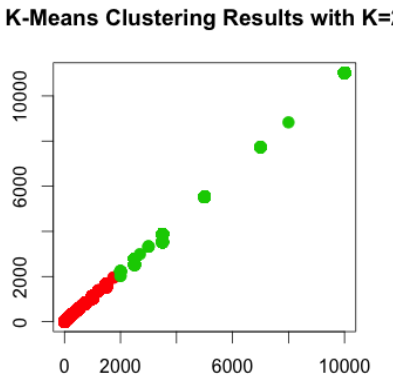
In terms of specific method, we can see that random forest without smote has the lowest error rate I, tree pruned has the lowest error rate II, and gam has the lowest error rate III. Therefore, when government want to reduce their error rate and to have better prediction performance, they have multiple choices. More specifically, if the government wants to predict data from all the sample, they choose random forest without smote to obtain the lowest error rate I. When government wants to predict accurately of those ticket that are non-complained, they might choose tree pruned for the lowest error rate II. Moreover, when government wants to predict accurately of the tickets that are compliant, gam is a better choice and smote will largely increase the prediction accuracy.

Thus, if the government is more interested in the relationship between predictors and the response, the parametric model is a better choice than the tree-ensemble methods. In general, the government can predict accurately of when the ticket is non-compliant especially when they have enough information of the variables including late fees, disposition and discount amount. And both parametric and nonparametric methods are reliable in predicting when the true response is non-compliant.

Clustering

To explore this data, in order to cluster data to predict compliance, we use k means and hierarchical clustering methods. However, there are two limitations in cluster methods.

1) Dimension of this data is very high. For example, the level of inspector_name is 173, violation_code is 235. Therefore, most of variables cannot use hierarchical method.
2) Since variables with low levels weakly related with compliance. For example, p-value of variable “zip” is 1, indicating that this variable is not significant at all. Therefore, it is not meaningful to explore the relationship between these variables and compliance.
3) In order to use k means, the dots are not spherical. Therefore, k-means is not an ideals way to cluster these points.



Therefore, clustering the data set makes little sense in the scenario.