

A Prediction Model Project in Python: Summary

Bernardo Di Chiara

February 21st 2020

Table of Contents

- Overview
- Input Datasets
- Input Variables
- Data Preparation Steps
- Data Cleaning Steps
- Exploratory Data Analysis
- Selected Variables for Prediction Models
- Data Preparation
- Used Evaluation Metrics
- Exploratory Prediction Model
- Used Models
- Prediction
- Conclusions
- References



Overview

- Based on a [Kaggle](#) challenge by the [Michigan Data Science Team](#) and the [Michigan Student Symposium for Interdisciplinary Statistical Sciences: Detroit Blight Ticket Compliance](#)
- Blight violation fines are issued by the city to individuals who allow their properties to remain in a deteriorated condition.
- Many of these fines remain unpaid.
- Goal: predict when a given blight ticket will be paid on time
- The work has been done by using Python.

Input Datasets

- **train.csv**: the training set
 - 250306 entries, including 90426 entries with null value in the target variable (159880 useful entries)
 - 34 variables
 - each entry corresponds to a single blight ticket.
 - timestamp cutoff at the end of year 2011 to avoid data leakage
 - target variable: compliance
 - True if the ticket was paid early, on time, or within one month of the hearing date
 - False if the ticket was paid later than one month after the hearing date or not at all
 - Null if the violator was found not responsible
- **test.csv**: the test set
 - 61001 entries
 - 27 variables (no target variable)
- **addresses.csv**: mapping from ticket id to violation addresses
 - 311307 entries (sum of train and test data sets entries)
 - 121769 unique addresses
- **latlons.csv**: mapping from violation addresses to lat/lon coordinates
 - 121769 entries

Input Variables (1/3)

Variable name	Description	Presence in dataset
ticket_id	unique identifier for blight tickets	train, test, addresses
agency_name	Agency that issued the ticket	train, test
inspector_name	name of the inspector that issued the ticket	train, test
violator_name	name of the person/organization that the ticket was issued to	train, test
violation_street_number, violation_street_name, violation_zip_code	address where the violation occurred	train, test
mailing_address_str_number, mailing_address_str_name, city, state, zip_code, non_us_str_code, country	mailing address of the violator	train, test
ticket_issued_date	date and time the ticket was issue	train, test
hearing_date	date and time the violator's hearing was scheduled	train, test
violation_code, violation_description	type of violation	train, test

Input Variables (2/3)

Variable name	Description	Presence in dataset
disposition	judgment and judgement type	train, test
fine_amount	violation fine amount, excluding fees	train, test
admin_fee	20 dollars fee assigned to responsible judgements	train, test
state_fee	10 dollars fee assigned to responsible judgments	train, test
late_fee	10% fee assigned to responsible judgments if the fine and fees imposed are not paid by the hearing date	train, test
discount_amount	discount applied, if any	train, test
clean_up_cost	clean-up or graffiti removal cost	train, test
judgment_amount	sum of all fines and fee	train, test
grafitti_status	flag for graffiti violations	train, test
payment_amount	amount paid, if any	train
payment_date	date payment was made, if it was received	train
payment_status	current payment status as of Feb 1 2017	train
balance_due	fines and fees still owed	train
collection_status	flag for payments in collections	train

Input Variables (3/3)

Variable name	Description	Presence in dataset
compliance	target variable	train
compliance_detail	more information on why each ticket was marked compliant or non-compliant	train
address	a combination of violation_street_number and violation_street_name	addresses, latlonss
lat	latitude corresponding to the violation address	latlonss
lon	longitude corresponding to the violation address	latlonss

- If the property owner complies before the hearing, the fine is withdrawn

Data Preparation Steps

- The `address` and the `latlons` data sets have been merged by using the `address` as a key and the result has been merged with both the `train` and the `test` datasets by using the `ticket_id` as a key.
- A new categorical variable (`violator_cat`) has been created to divide violators in 4 classes based on the amount of violations by using the `violator_name` variable as input.
 - 45.652 out of 119.992 violators have had more than 1 accident
 - 1864 of them have had more that 10 accidents
 - 19 of them have had more that 100 accidents
- Other two categorical variables (`late_fee` and `discount`) have been created to indicate whether there has been a late fee and whether there has been a discount (since the late fee is always a fixed percentage of the fine amount and the discount, when applied, it is within a very narrow percentage range).
- Year and month have been extracted from the `ticket_issued_date` into two separate variables (`year` and `month`).

Data Cleansing Steps

- Three entries with wrong data in the ticket issued date have been eliminated from the `train` dataset.

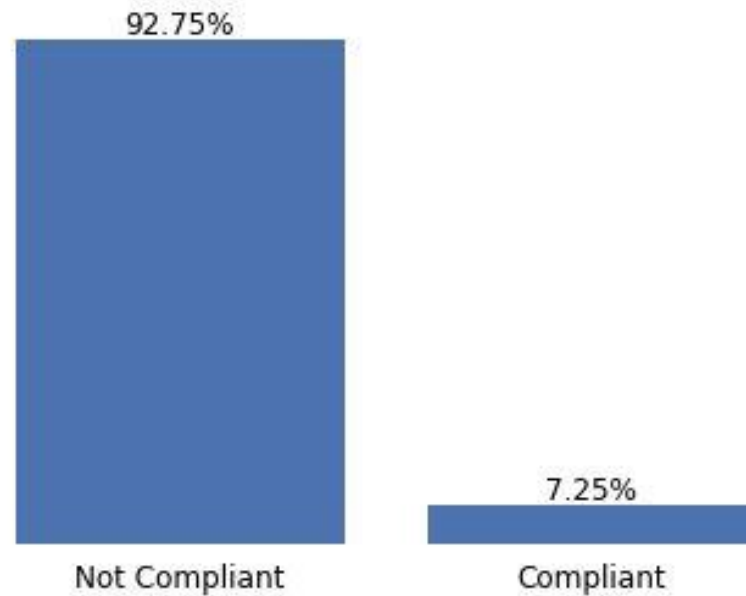
```
1938-10-09 15:30:00  
1963-03-17 10:00:00  
1988-05-06 20:00:00
```

- The entries with finite value in the compliance (entries related to responsible violators) have been selected from the `train` dataset.
- The few entries with null values on the coordinates and in the violator category variable have been filled by using the mean for the numerical variables and the most frequent value for the categorical variable.
- It has been verified automatically that there were no miss-spelled addresses in the additional files.

Exploratory Data Analysis

Target value: Compliance

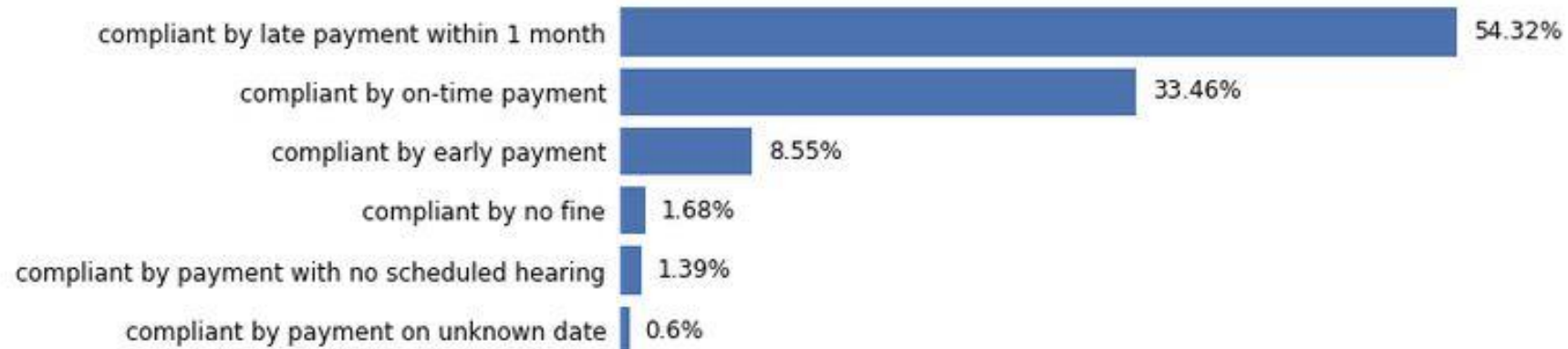
Distribution of values for the variable compliance in percentage



Exploratory Data Analysis

Compliance details

Distribution of values for the variable compliance_detail among compliant violators



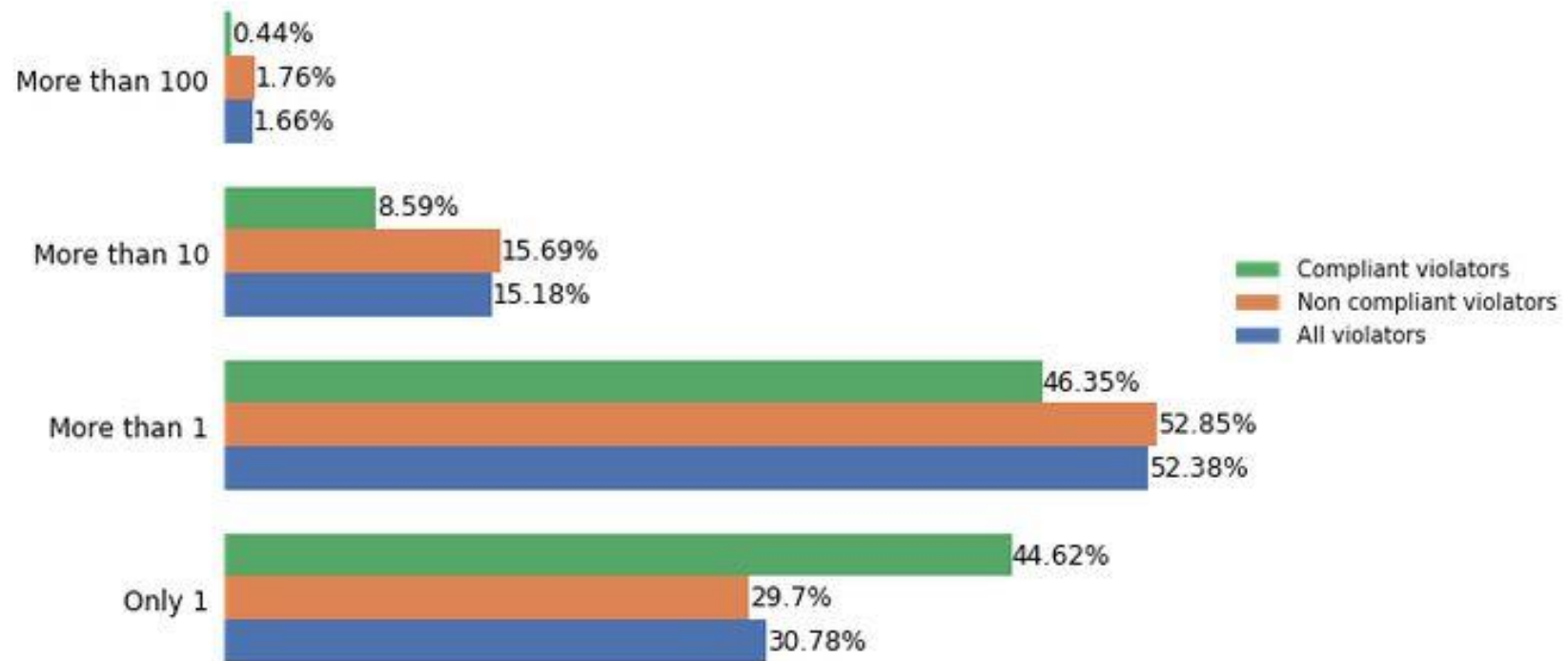
Distribution of values for the variable compliance_detail among non compliant violators



Exploratory Data Analysis

Violator category (1/2)

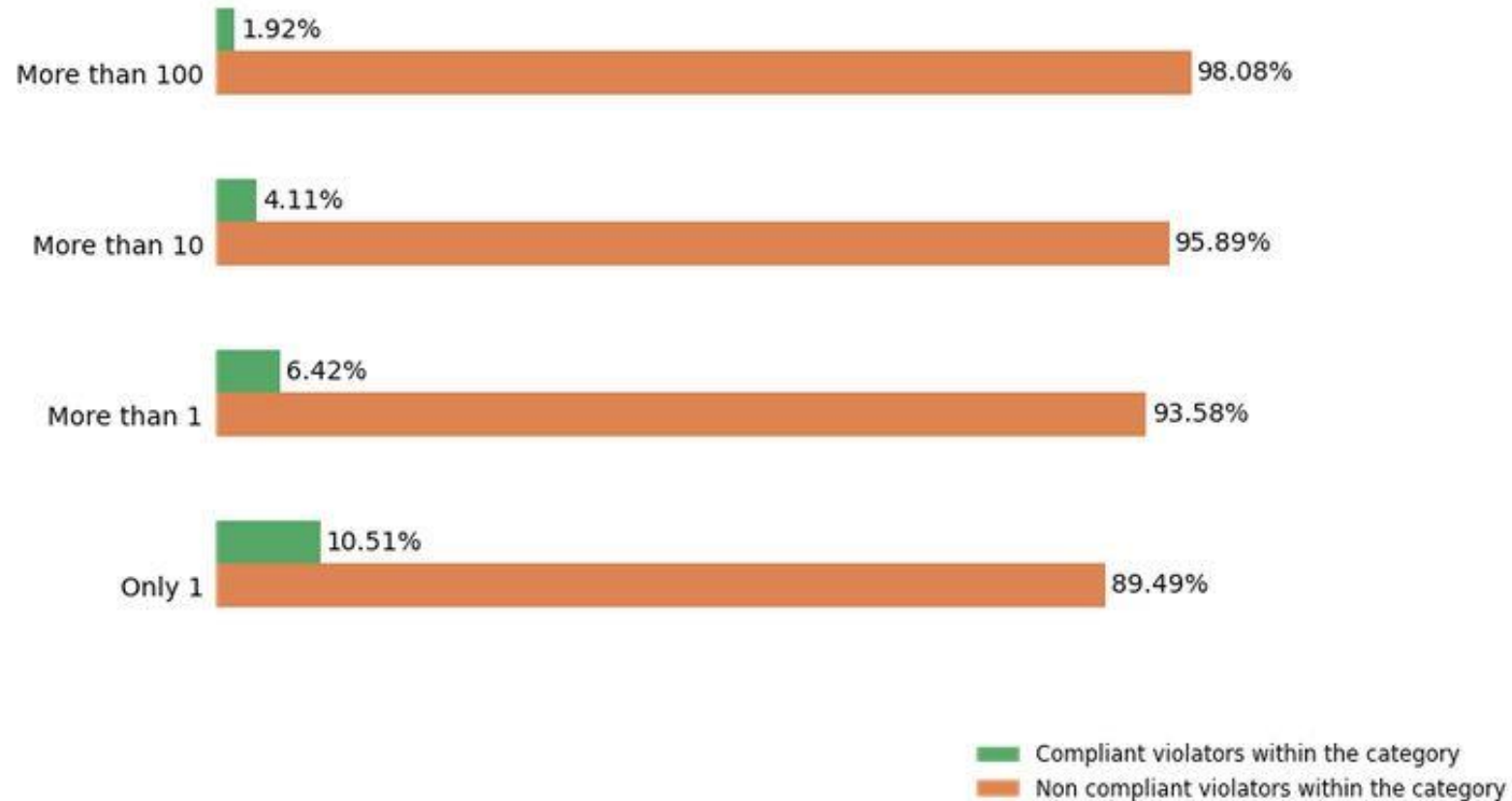
Distribution of values for the variable violator_cat in percentage for different groups of violators



Exploratory Data Analysis

Violator category (2/2)

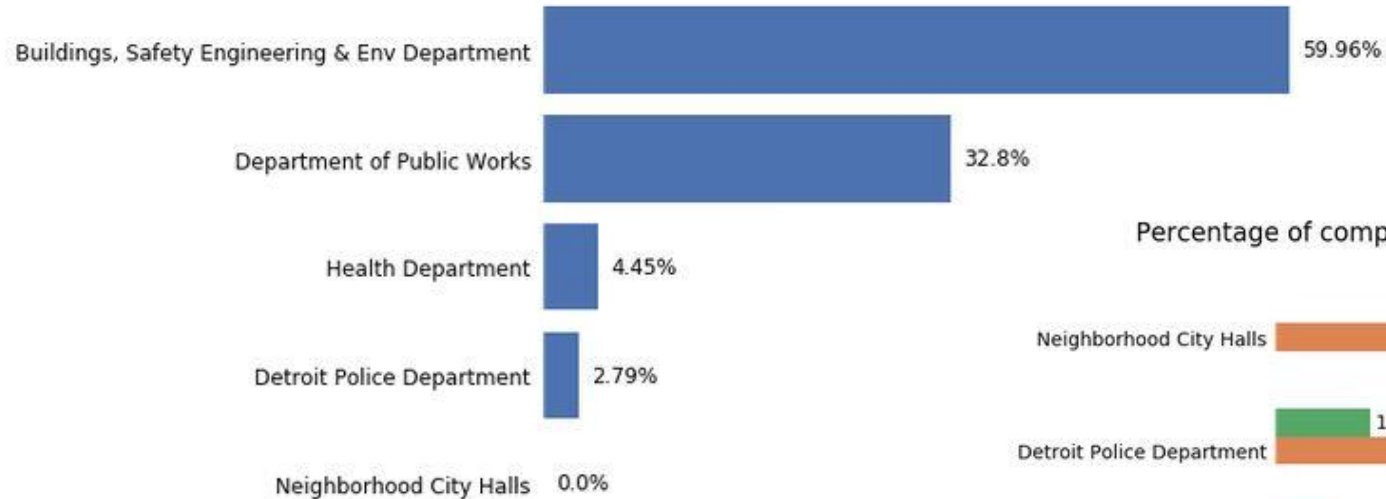
Percentage of compliant and non compliant responsible violators for each violator category



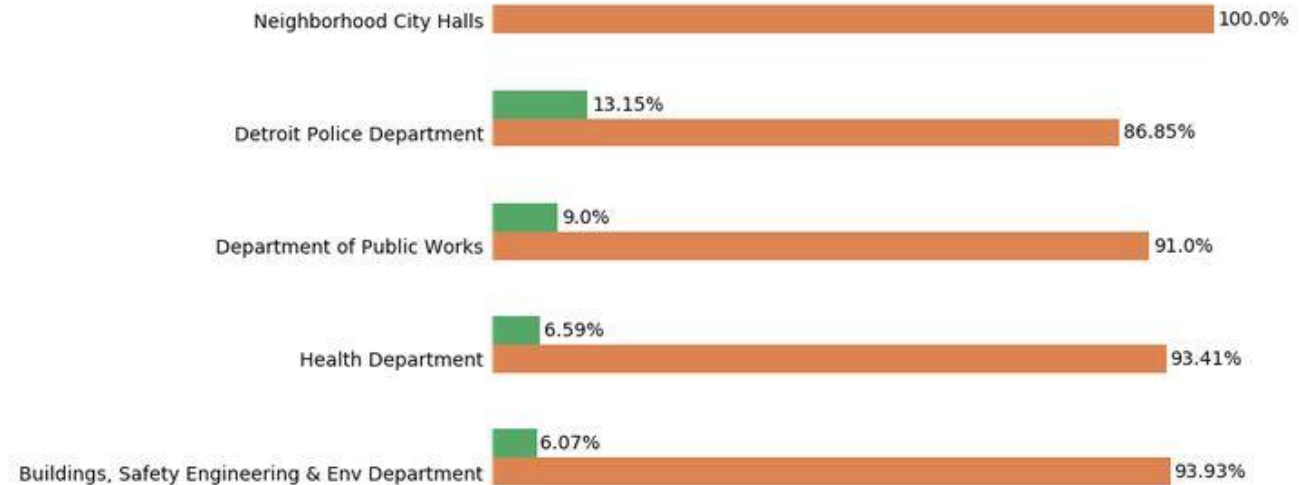
Exploratory Data Analysis

Agency name

Amount of tickets to responsible violators for each agency in percentage



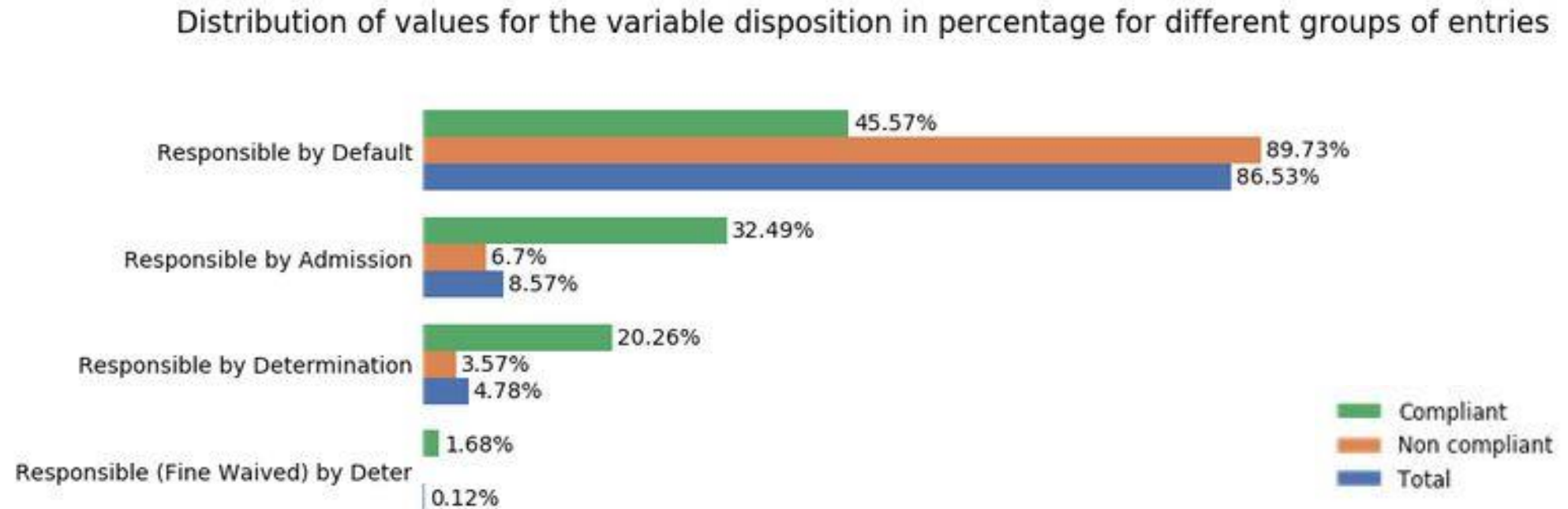
Percentage of compliant and non compliant responsible violators for each agency



Compliant violators within the category
Non compliant within the category

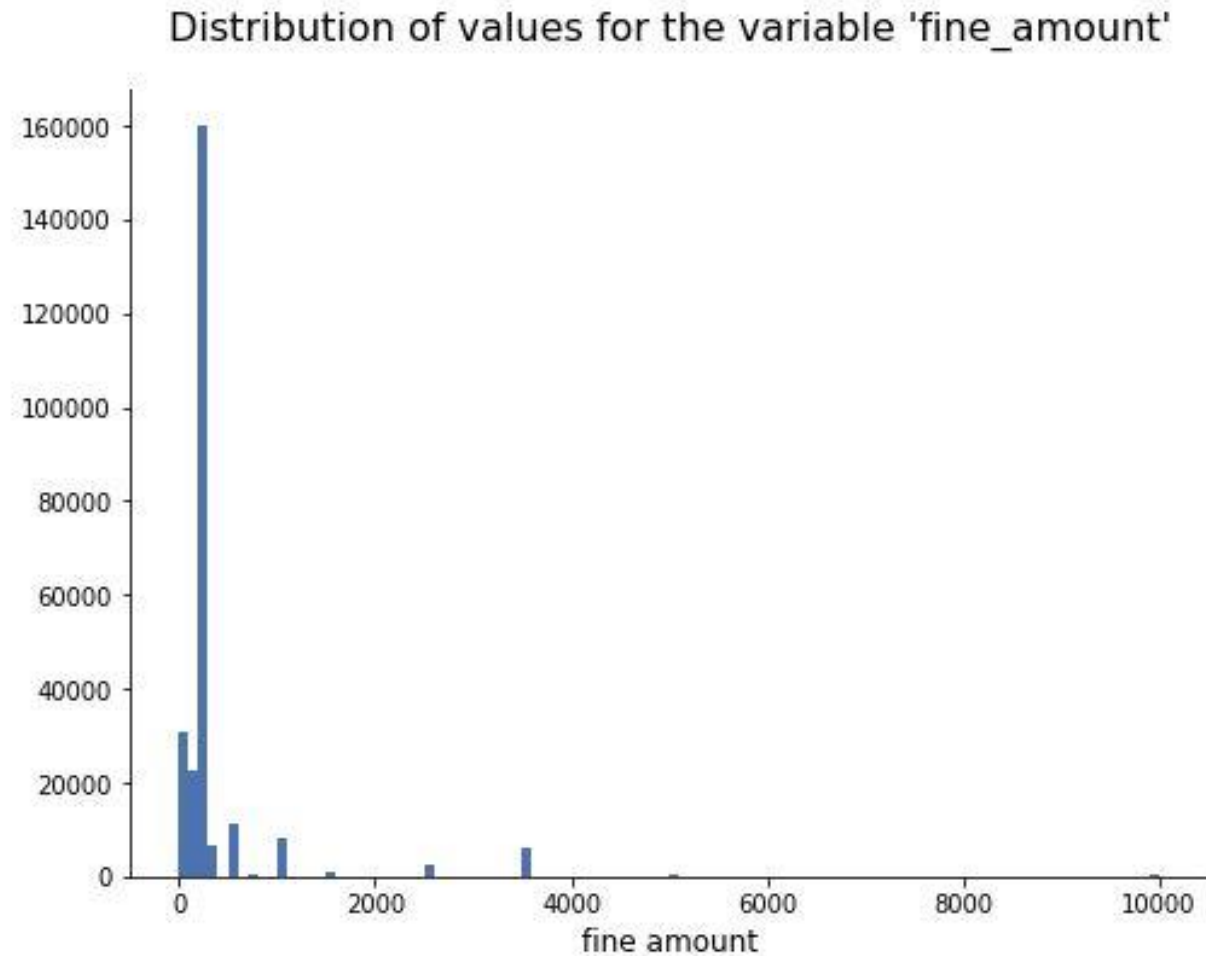
Exploratory Data Analysis

Disposition



Exploratory Data Analysis

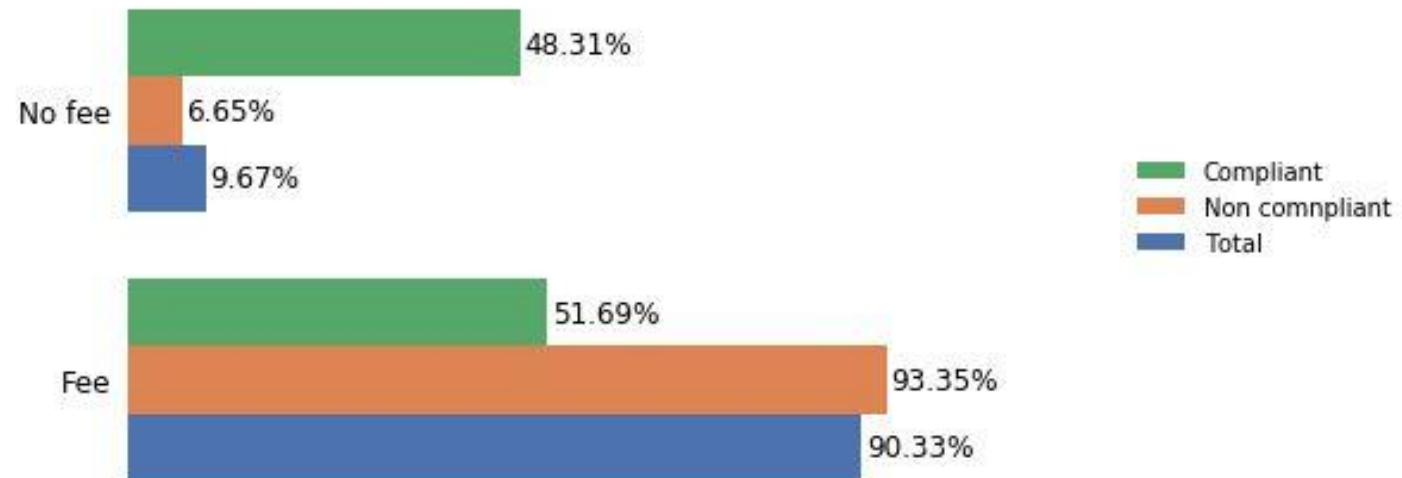
Fine amount



Exploratory Data Analysis

Late fee

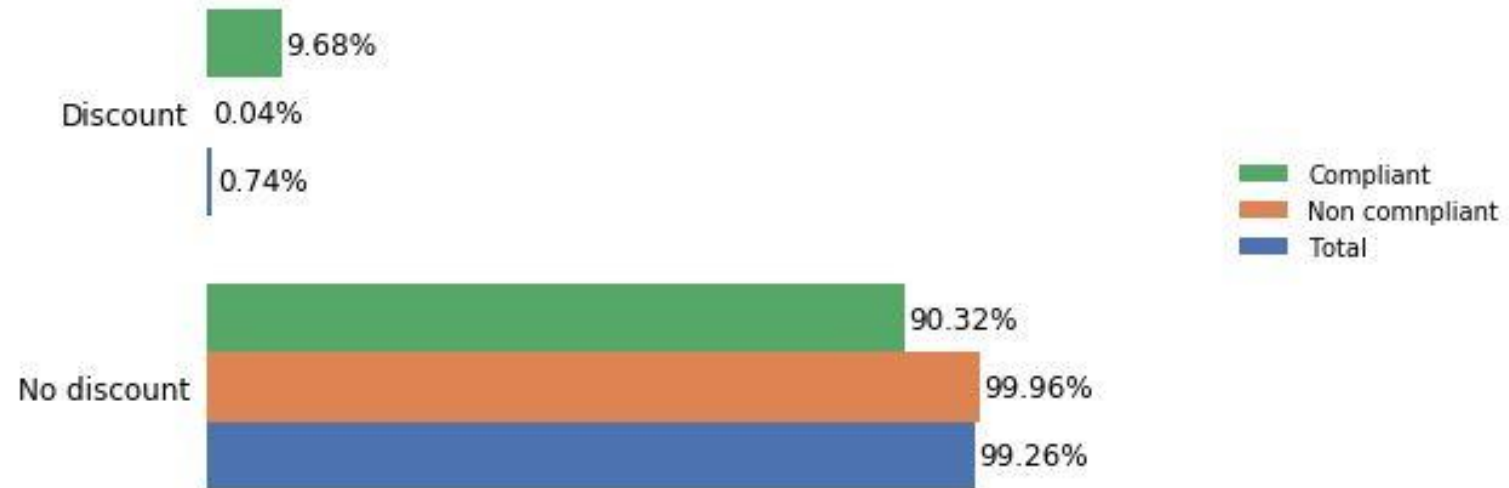
Distribution of values for the variable late_fee in percentage for different groups of entries



Exploratory Data Analysis

Discount

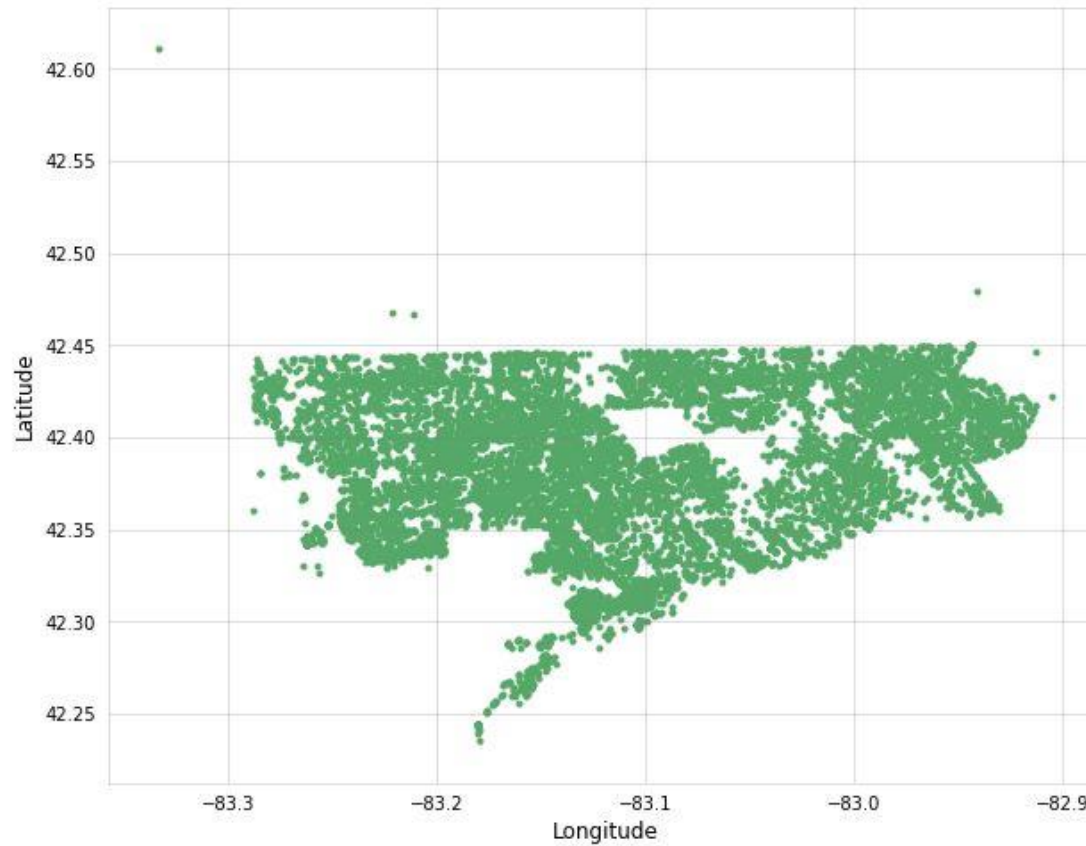
Distribution of values for the variable discount in percentage for different groups of entries



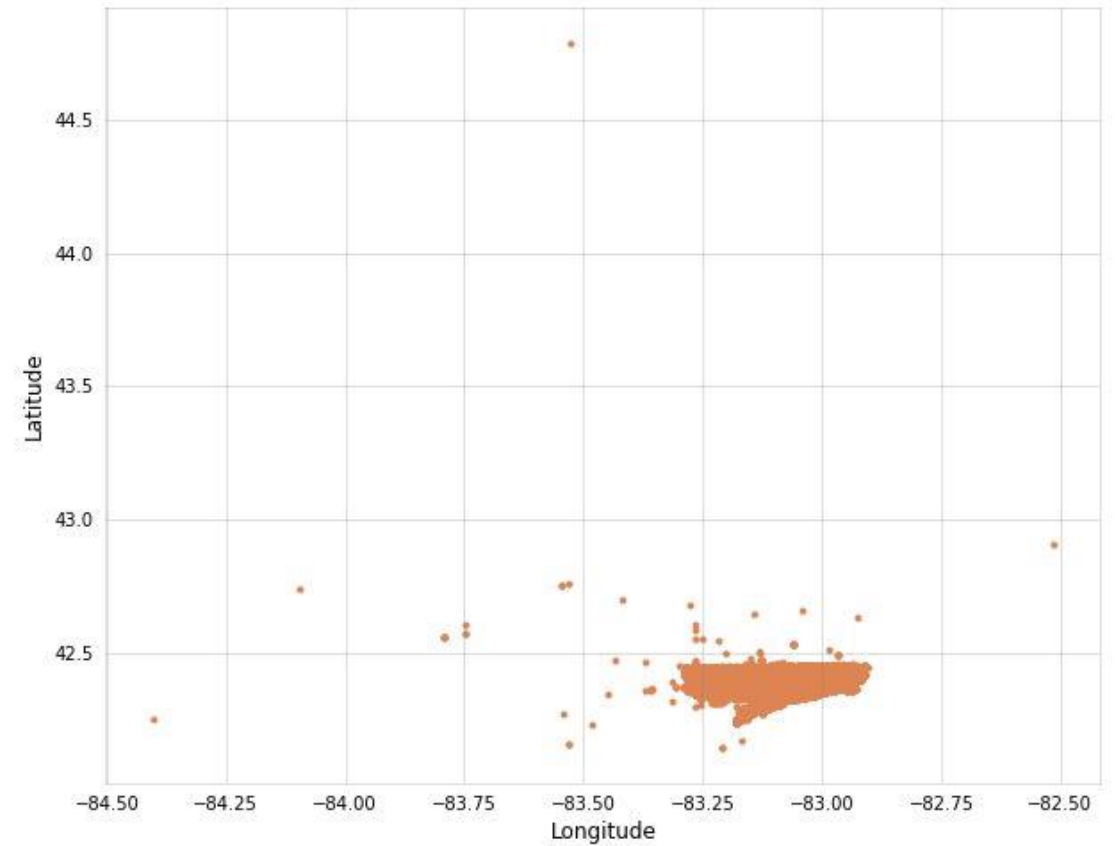
Exploratory Data Analysis

Location

Latitude vs Longitude for Compliant Violators



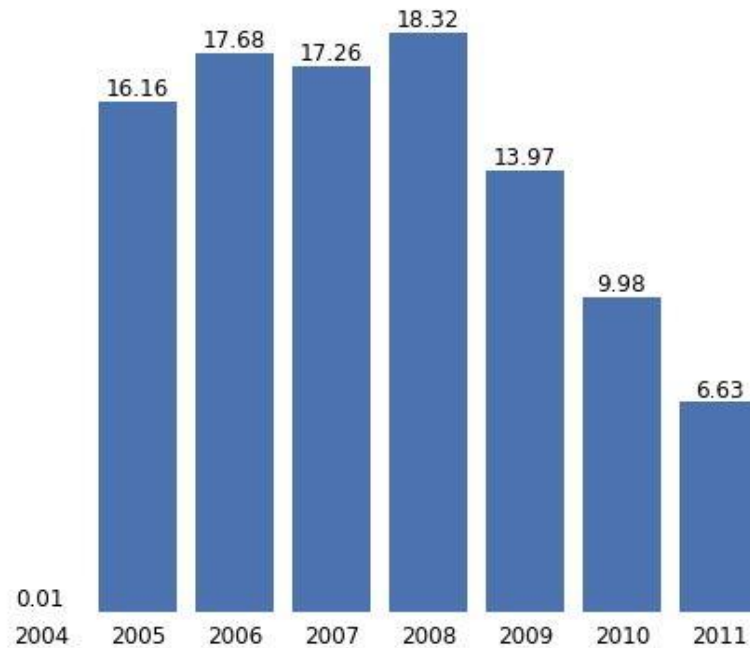
Latitude vs Longitude for non Compliant Violators



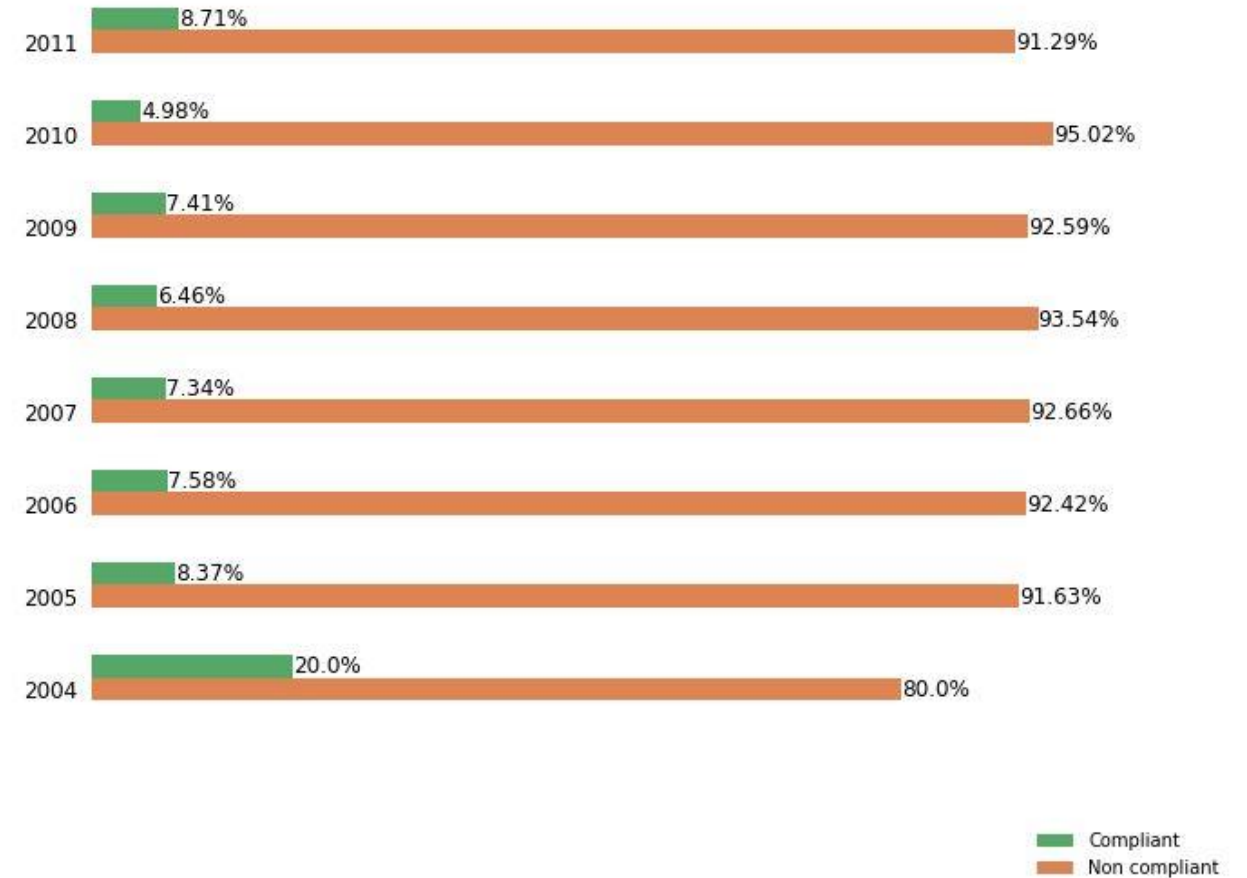
Exploratory Data Analysis

Year

Percentage of tickets to responsible violators for each year



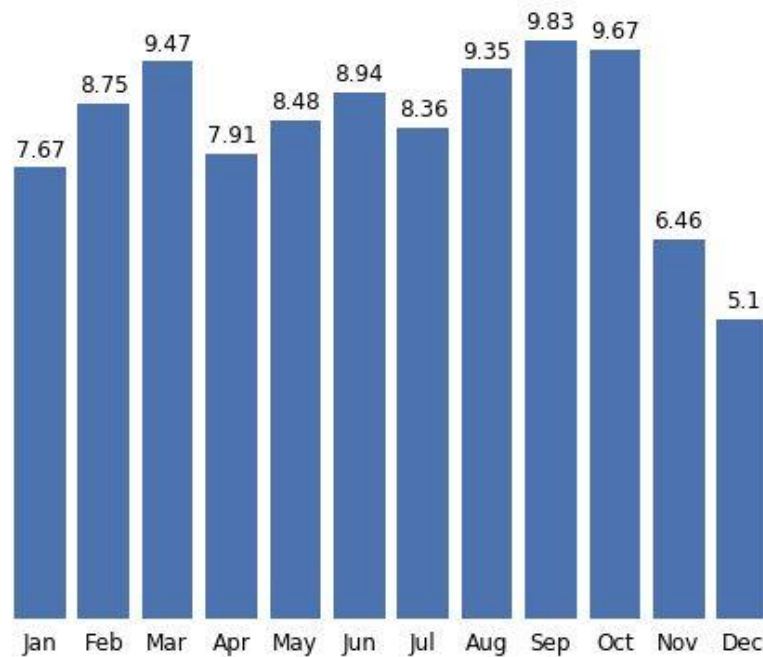
Percentage of compliant and non compliant responsible violators for each year



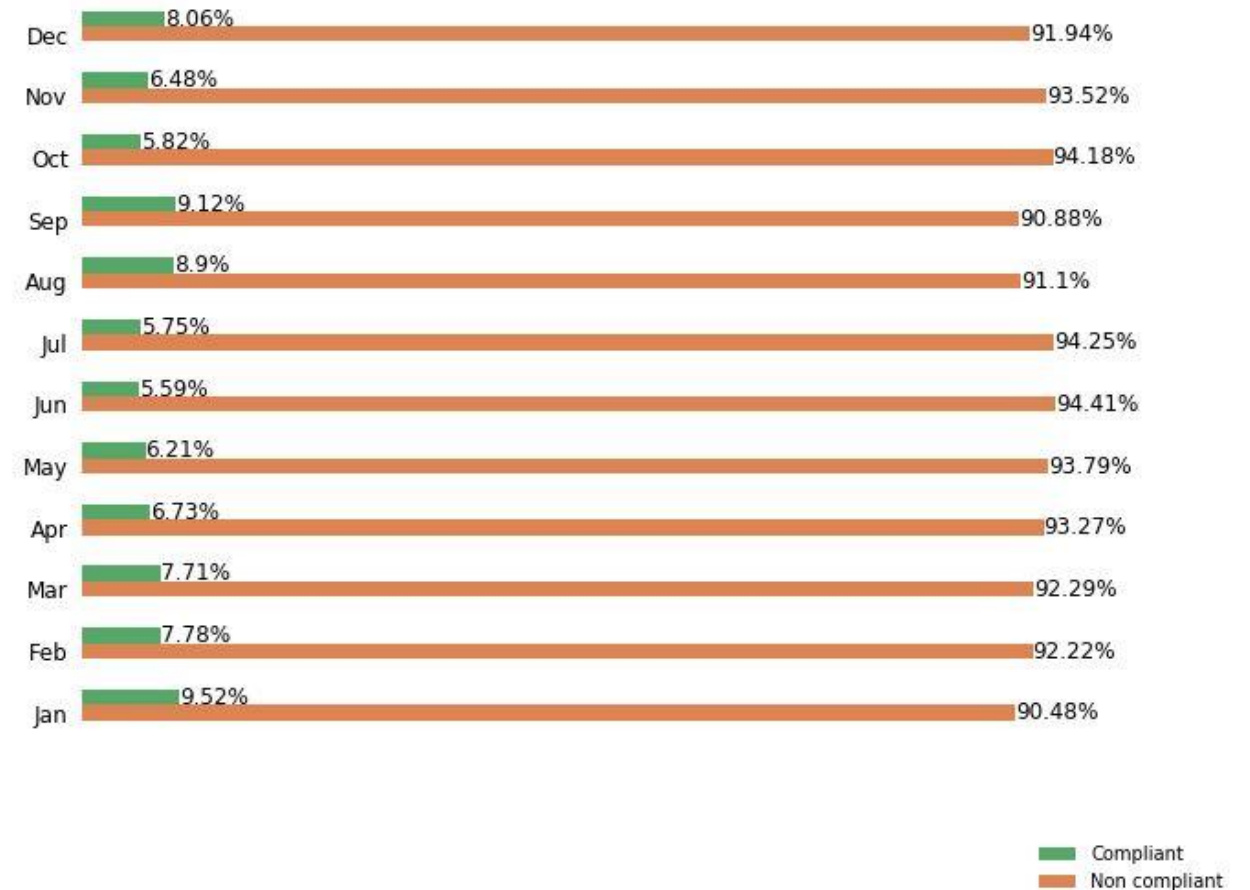
Exploratory Data Analysis

Month

Percentage of tickets to responsible violators for each month



Percentage of compliant and non compliant responsible violators for each month



Selected Variables for Prediction Models

Variable name	Variable type	Details
ticket_id	numerical	used only as a key for alignment to the test dataset
agency_name	categorical (5 values)	
year	numerical	created feature
month	numerical	created feature
violator_cat	categorical (4 values)	created feature
lat	numerical	
lon	numerical	
disposition	categorical (13 values)	
fine_amount	numerical	
late_fee	categorical (2 values)	created feature
discount	categorical (2 values)	created feature
compliance	numerical	target value

Data Preparation

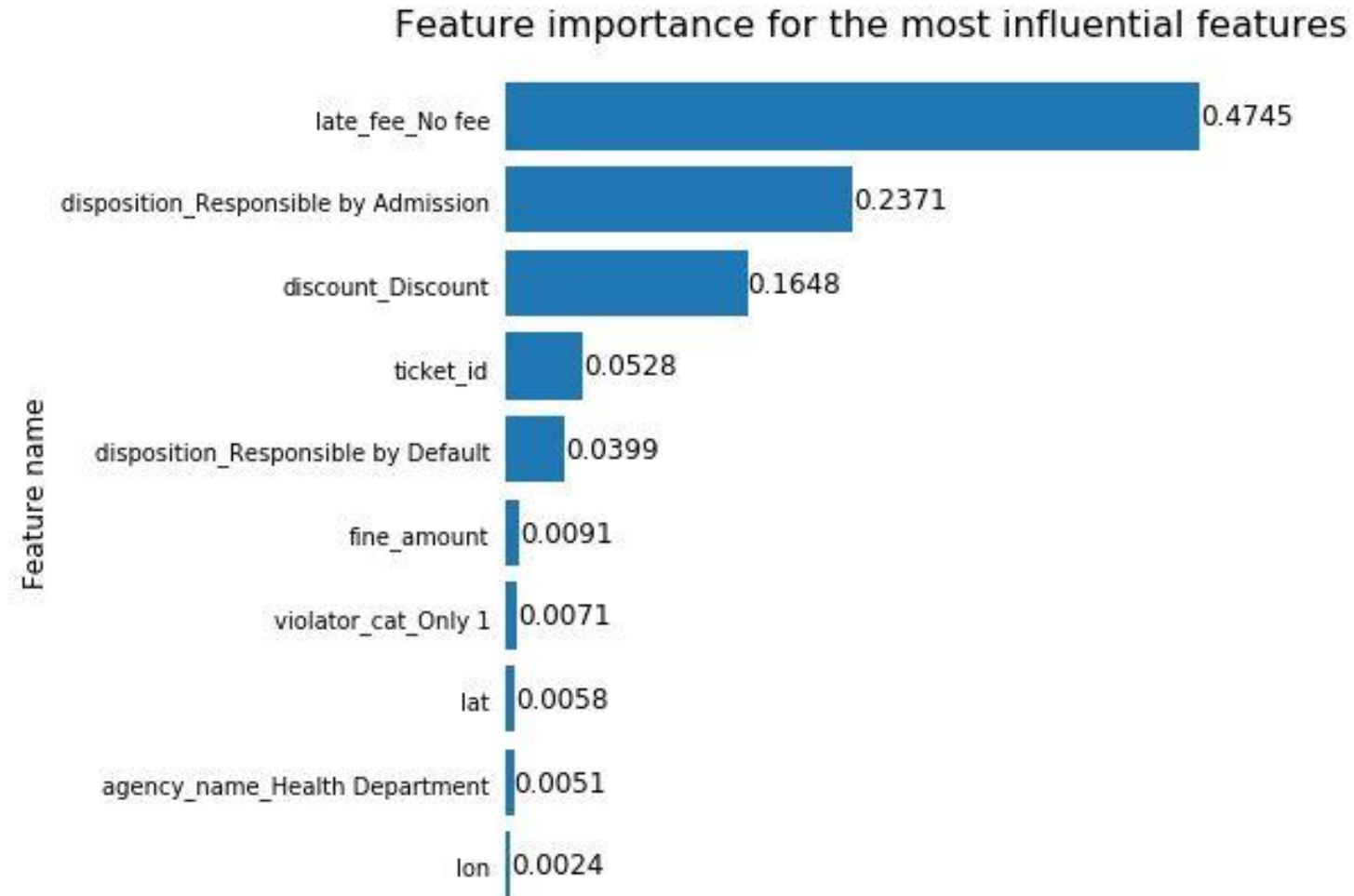
- The category names in the categorical variables have been aligned in the train and the test data sets.
 - The categorical variables have been converted into indicator variables.
 - Modifications have been done to the test dataset in order to align it with the modifications done to the training data set.
 - A validation data set has been extracted from the train dataset to be used for model evaluation.
 - The features have been normalized in a separate data set to be used with liner models.
- A dataset with 32 features, containing sparse data

Used Evaluation Metrics

- Due to the presence of many unbalanced classes, accuracy was not a useful evaluation metrics.
- The AUC (area under the ROC curve) has been used.
- During parameter tuning, an **overfitting coefficient** has been defined in order to limit overfitting.
 - $(\text{AUC on train set} - \text{AUC on validation set}) / (\text{AUC on validation set}) * 100$
- The **overfitting coefficient** has been kept below 1%.

Exploratory Prediction Model

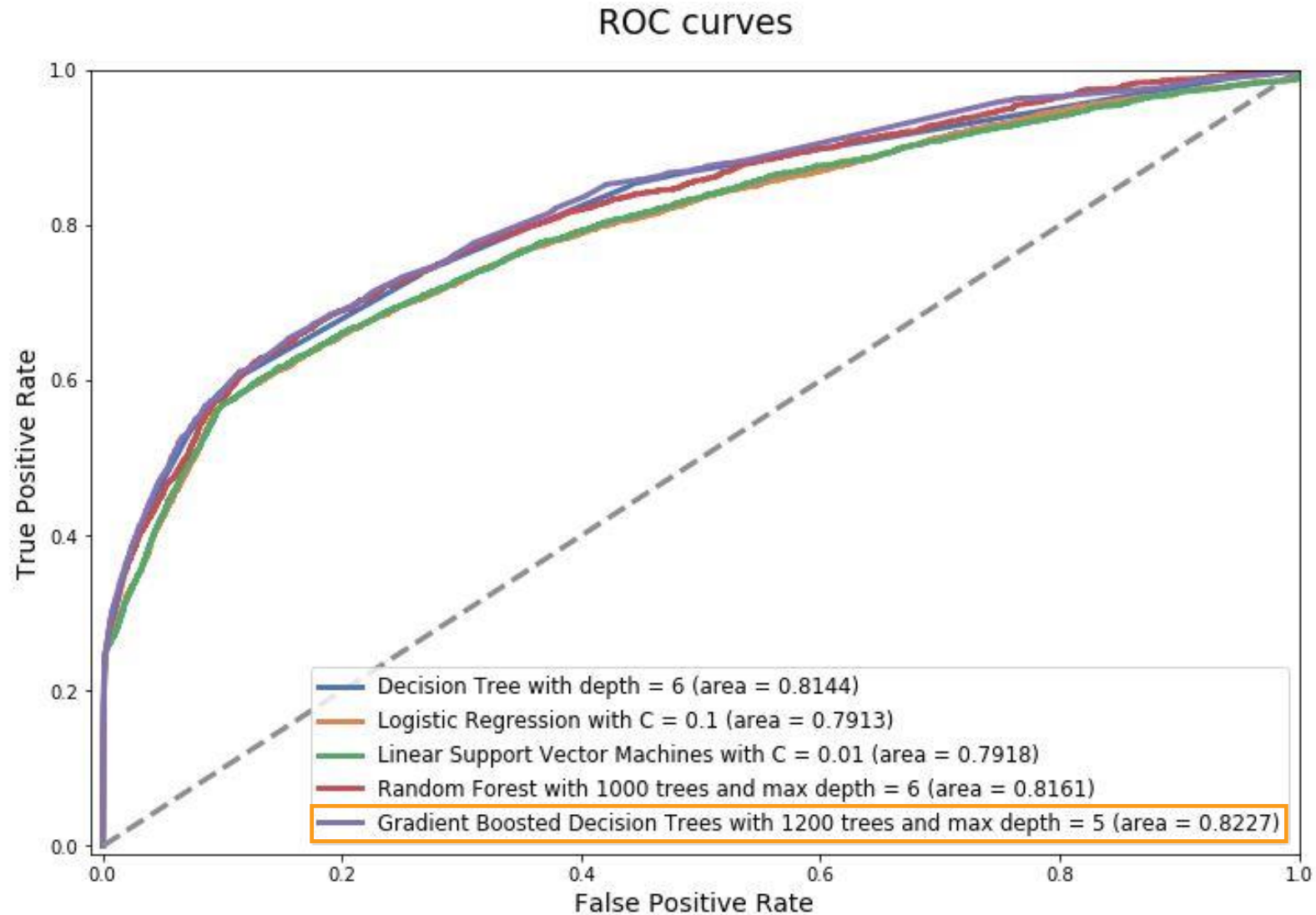
Decision Tree



Used Models (1/2)

Model	Parameters Tuning	AUC on train set	AUC on validation set	Overfitting coefficient
Decision Tree	max_depth=6 min_samples_split=30 min_samples_leaf=15	0.8195	0.8144	0.63%
Logistic Regression	C=0.1 solver="liblinear"	0.7963	0.7913	0.64%
Linear Support Vector Machines	C=0.01	0.7977	0.7918	0.74
Random Forest	n_estimators=1000 max_depth=6 min_samples_split=24 min_samples_leaf=12	0.8210	0.8161	0.60
Gradient Boosted Decision Trees	n_estimators=1200 max_depth=5 min_samples_split=20 min_samples_leaf=10 learning_rate=0.0015	0.8287	0.8227	0.72%

Used Models (2/2)




Prediction

- The gradient boosted decision tree model has been applied to the test set.
- The probability that the blight ticket will be paid at latest within one month of the hearing date has been calculated.
- The resulting AUC on the test set has been **0.8221 (*)**.
 - The result is within the top 3 scores!

(*) obtained through a third party company

<https://www.kaggle.com/c/detroit-blight-ticket-compliance/leaderboard>



CITY OF
Detroit

Detroit Blight Ticket Compliance

Help end blight in Detroit.







24 teams · 3 years ago

OverviewDataDiscussionLeaderboardRules

Public LeaderboardPrivate Leaderboard

The private leaderboard is calculated with approximately 50% of the test data.

This competition has completed. This leaderboard reflects the final standings.

#	Δpub	Team Name	Notebook	Team Members	Score ?
1	—	Jared Webb			0.83392
2	—	raskolnikov			0.82256
3	—	KS		   	0.82147

Conclusions

- An extensive **data analysis** has been performed after careful **data preparation and cleansing**.
- This has allowed to check extreme values and the consequent need for normalization, to help in the selection of the potential predictors (features) and to give hints about the prediction methods that would be most likely successful.
- This analysis has also helped to confirm that the selected features (like discount and late fee) would not introduce risks of data leakage.
- A **Decision Tree** model has been used as a preliminary model. Two linear classification models (Logistic Regression and **Linear Support Vector Machines**) as and two ensemble models (**Random Forest** and **Gradient Boosted Decision Tree**) have been fit as well.
- The models have been **evaluated** by using the **AUC** (area under the ROC curve) and the parameters have been **fine tuned** by keeping the **AUC value on the train set less than 1% bigger** than the AUC value on the validation set.
- The **Gradient Boosted Decision Tree** model has been selected.
- The result has been a AUC value of **0.8221** on the test set, not bad by considering that it would have been **among the top 3 scores in the final competition!**

References

- A_Prediction_Model_in_Python_Complete_Code
 - This file contains the complete code used to dump, prepare, clean, analyze and visualize data as well as the code used to fit and evaluate the prediction models and to make the prediction
<https://github.com/BerniHacker/CV>
- Kaggle: Detroit Blight Ticket Compliance
<https://www.kaggle.com/c/detroit-blight-ticket-compliance/overview>