BLIGHT TICKET COMPLIANCE PREDICTION

ARATHY AJITH (C0790836)
JIBIN JOHN (C0796674)
NIDHI PARMAR (C0796680)
TARANPREET KAUR (C0793771)
RINCY JOSE (C0790298)

AIMT FALL 2020, LAMBTON COLLEGE
AML 2203 - ADVANCED PYTHON AI AND ML TOOLS
VAHID HADAVI
APRIL 10, 2021

Table of Contents

1. Introduction **2**

2. Dataset **2**

3. Data Pre-processing **3**

3.1. Missing Value Handling **5**

4. Data Visualization **6**

5. Feature Importance **9**

6. Feature Selection **10**

7. ML Modelling **10**

8. Results and Conclusion **10**

9. Reference **11**

1. Introduction

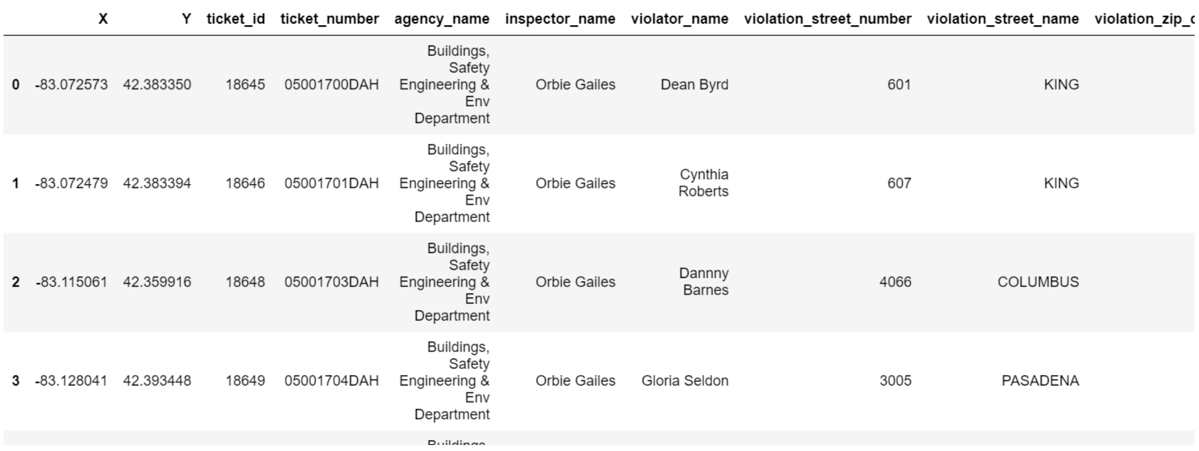
Blight, or properties that have not been adequately maintained, is a common and costly issue in Detroit. Blight violation notices, also known as blight tickets, were first issued by Detroit's city in 2005 to allow residents to maintain their homes in good repair or contribute to the city's renewal efforts. Landowners who have violated City of Detroit ordinances regulating how property owners must preserve the exterior of their property have been issued Blight Violation Notices (BVN) or Blight Tickets, and many of these fines go unpaid. The city government requires measures to increase blight ticket compliance. Enforcing unpaid blight fines is an expensive and time-consuming operation.

Through this project, the teams' primary goal is to explore what factors make a citizen comply with a blight ticket, thereby predicting whether a specific blight ticket would be paid on time.

2. Dataset

The dataset was retrieved from the City of Detroit Open Data Portal. This data is freely accessible in human-readable formats, and it is formatted according to national technical requirements to promote data visibility and reuse. The dataset contains 501,689 records and 41 features as of March 23, 2021. The dataset keeps on updating on weekly.

The target variable was manually created as a part of the pre-processing step and was named ‘Compliance’, highlighting whether the ticket holder is compliant or not.



	X	Y	ticket_id	ticket_number	agency_name	inspector_name	violator_name	violation_street_number	violation_street_name	violation_zip_c
0	-83.072573	42.383350	18645	05001700DAH	Buildings, Safety Engineering & Env Department	Orbie Gailes	Dean Byrd	601	KING	
1	-83.072479	42.383394	18646	05001701DAH	Buildings, Safety Engineering & Env Department	Orbie Gailes	Cynthia Roberts	607	KING	
2	-83.115061	42.359916	18648	05001703DAH	Buildings, Safety Engineering & Env Department	Orbie Gailes	Dannny Barnes	4066	COLUMBUS	
3	-83.128041	42.393448	18649	05001704DAH	Buildings, Safety Engineering & Env Department	Orbie Gailes	Gloria Seldon	3005	PASADENA	

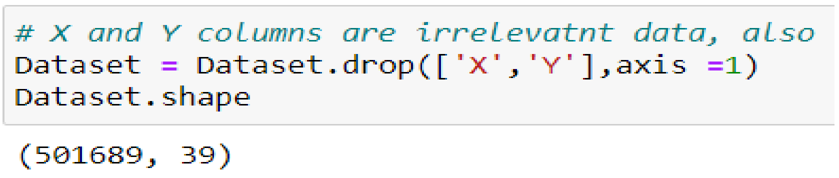
Figure 1: Overview of Blight Dataset

The major columns in the dataset are:

- ticket_id - the unique identifier for tickets
- agency_name - the agency that issued the ticket
- inspector_name - Name of the inspector who issued the tickets
- violator_name - Name of the person/organization to which tickets was issued to
- violation_street_number, violation_street_name, violation_zip_code - Address details where the violation has occurred
- mailing_address_str_number, city, state, mailing_address_str_name, zip_code, non_us_str_code, country - address of the violator
- ticket_issued_date – the date and time in which the tickets were issued
- hearing_date – The date and time the violator's hearing was scheduled
- violation_code, violation_description – the type of violation
- disposition - Judgment and judgement type
- fine_amount - Violation fine amount, excluding fees
- admin_fee - fee assigned for responsible judgments
- payment_amount - Amount paid if any
- payment_date – the date payment was made
- payment_status - the present payment status as of February 1, 2017
- balance_due – the balance fines and fees still owed
- collection_status - Flag for payments in collections

3. Data Pre-processing

For analysis, the dataset went through pre-processing so as to make it in a usable format. There are a lot of redundant columns that either have no data or have no useful information. Hence the first step in pre-processing is removing those irrelevant columns from the dataset. The X and Y columns represent the latitude and longitude of the property and impart no relevance to the model prediction. Hence, X and Y columns are removed from the dataset.



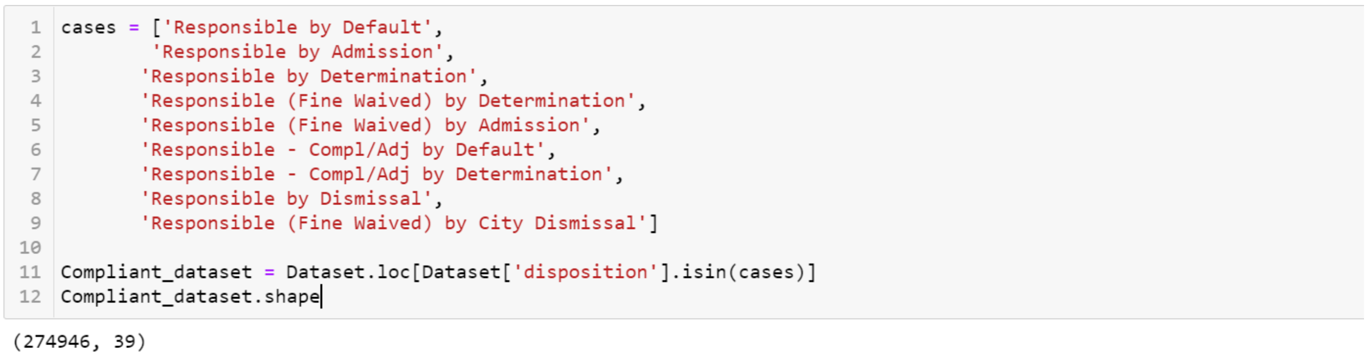
```
# X and Y columns are irrelevatnt data, also
Dataset = Dataset.drop(['X','Y'],axis =1)
Dataset.shape
```

```
(501689, 39)
```

Figure 2: Removing columns from dataset

The project requires to include people who have been found guilty or have to pay a fine, but the dataset consists of records for which a person is not guilty or doesn't have to pay any fine. This particular characteristic is determined by the 'disposition' column. The values which hold the term 'responsible' highlight the scenarios mentioned earlier. A few terms such as 'by,' 'by Default' and 'Responsible by' seemed ambiguous and were further analysed.

While evaluating the disposition term 'by,' it can be explicitly mentioned that all cases with a 'by' disposition have no judgement date, making the records vague for further analysis. Therefore, the dataset needs to be filtered for only those who have disposition 'responsible by.'



```
cases = ['Responsible by Default',
         'Responsible by Admission',
        'Responsible by Determination',
        'Responsible (Fine Waived) by Determination',
        'Responsible (Fine Waived) by Admission',
        'Responsible - Compl/Adj by Default',
        'Responsible - Compl/Adj by Determination',
        'Responsible by Dismissal',
        'Responsible (Fine Waived) by City Dismissal']

Compliant_dataset = Dataset.loc[Dataset['disposition'].isin(cases)]
Compliant_dataset.shape
```

```
(274946, 39)
```

Figure 3: Filtering disposition for responsible cases

The next step is to determine whether or not an individual is compliant. For anyone to be compliant, the following conditions must be met:

- The person pays a fine within six months after the judgement date
- The person pays the total amount for fine
- The person pays a fine before judgement (By own admission)

3.1. Missing Value Handling

The next step is missing value handling, which is crucial in this dataset as it contains many columns with more than ~75% missing values. The bar chart representation of missing values is given below.

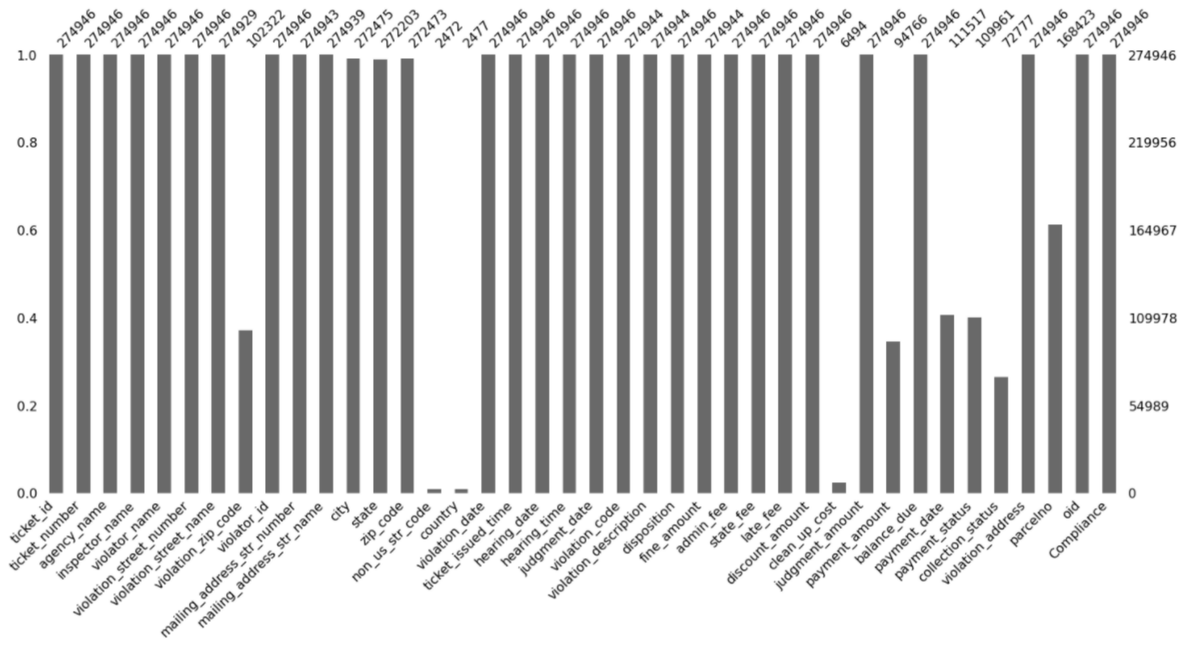
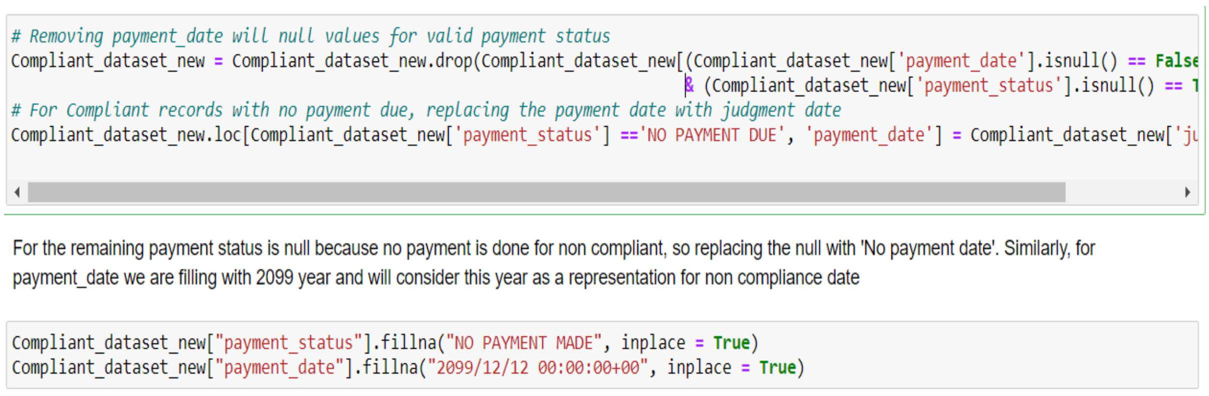


Figure 4: Missing value count

From the graph, it is apparent that the columns ticket_id','ticket_number',’violation_zip_code’, ‘non_us_str_code’, ‘country’, ‘payment_amount’, ‘collection_status’, ‘parcelno’, ‘clean_up_cost’ and 'oid' has a lot of missing values and didn’t contribute to any significant information which couldn’t be retrieved from other features, hence were removed from the dataset.

The missing values in columns 'payment_status' and 'payment_date' were handled by imputing the values 'NO PAYMENT MADE' and '2099/12/12 00:00:00+00' for existing non-compliant records. There were records for which ‘payment_status’ showed no dues, but ‘payment_date’ was null for which ‘judgment_date’ was used to impute.



```
# Removing payment_date will null values for valid payment status
Compliant_dataset_new = Compliant_dataset_new.drop(Compliant_dataset_new[(Compliant_dataset_new['payment_date'].isnull() == False
                                                                          & (Compliant_dataset_new['payment_status'].isnull() == T
# For Compliant records with no payment due, replacing the payment date with judgment date
Compliant_dataset_new.loc[Compliant_dataset_new['payment_status'] =='NO PAYMENT DUE', 'payment_date'] = Compliant_dataset_new['ju
```

For the remaining payment status is null because no payment is done for non compliant, so replacing the null with 'No payment date'. Similarly, for payment_date we are filling with 2099 year and will consider this year as a representation for non compliance date

```
Compliant_dataset_new["payment_status"].fillna("NO PAYMENT MADE", inplace = True)
Compliant_dataset_new["payment_date"].fillna("2099/12/12 00:00:00+00", inplace = True)
```

Figure 5: Imputation of few columns

4. Data Visualization

The visualization part focuses mainly to capture the relationship among the features and to figure out the trends and patterns in this dataset.

Few noted observations were made, which are as follows:

- Below figure highlights the time period a non-compliant person takes to pay his/her fine. It was noted that after a period of 26 months, the person could be considered a permanent defaulter.

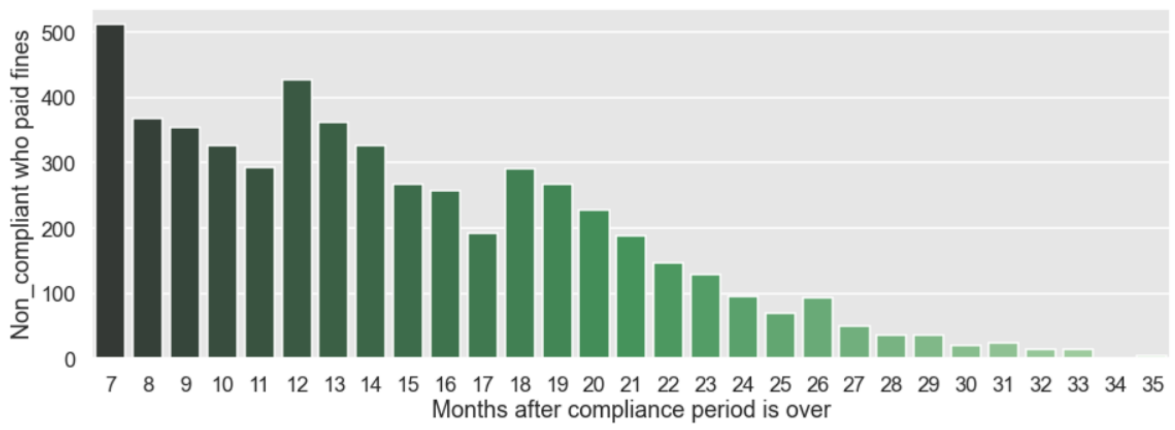


Figure 6: Months taken by non-compliant to pay fines

- The compliance was noted for those tickets for which discounts were given.

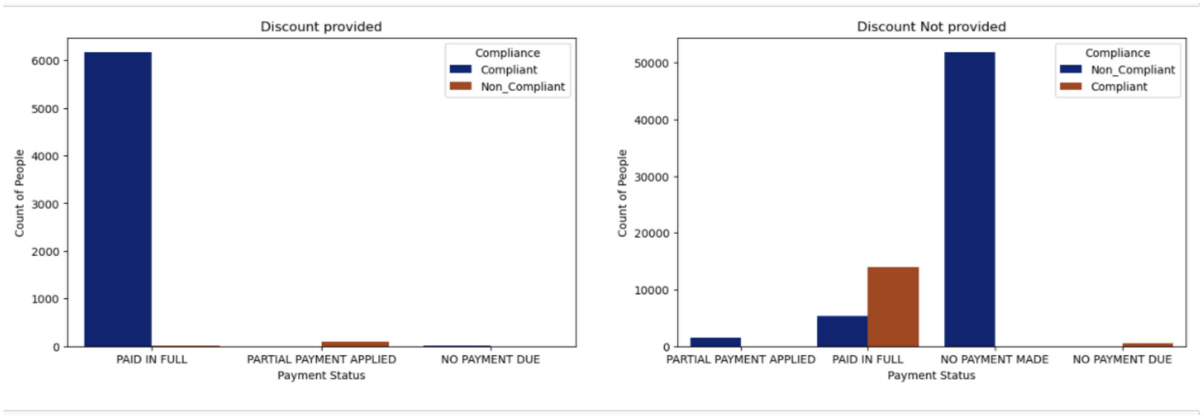


Figure 7: Discount effect on compliance

- The higher the fine, the chances of being non-complaint increased, whereas if the discount amount is high, the chances of being compliant also increase.

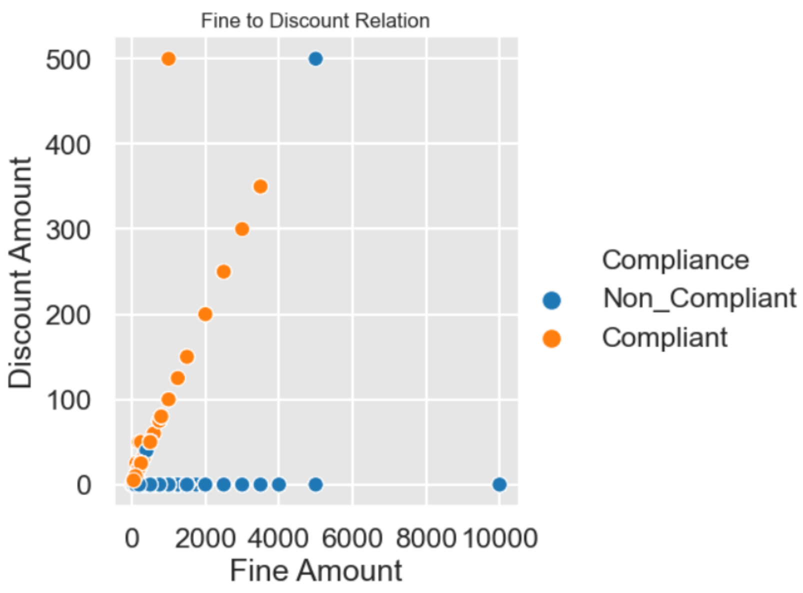


Figure 8: Discount to fine ratio

Compliant and non-compliant blight ticket holders of the five most cities were plotted as well. The compliance rate of the top 5 cities is found by taking the count of compliant and non-compliant incidents. From the graph, it is evident that all cities have high non-compliance rates as compared to compliance rates.

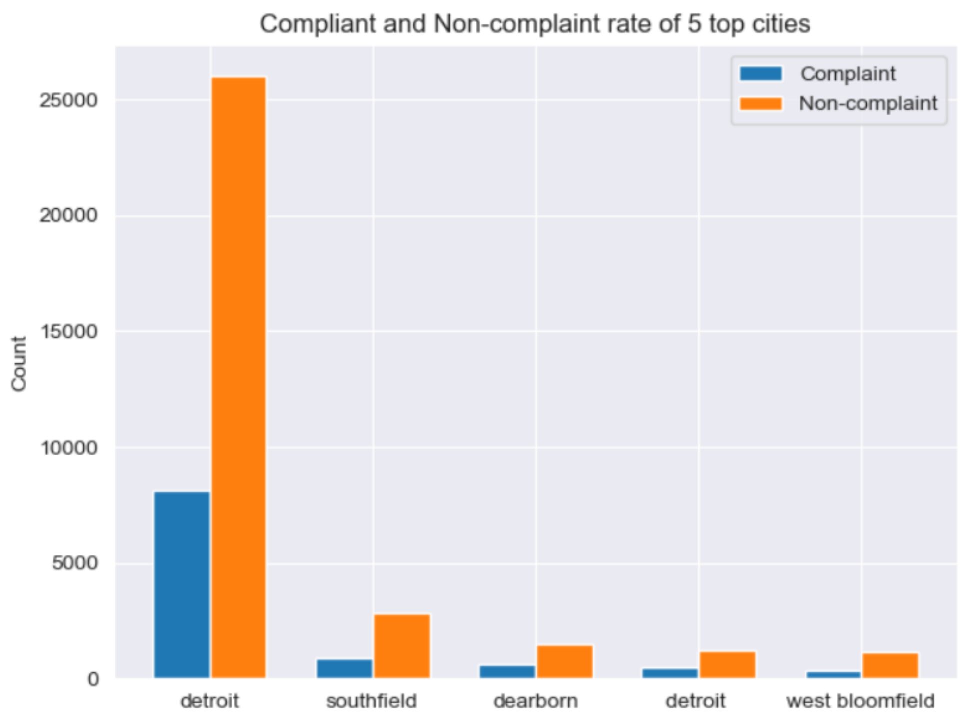


Figure 9: Compliance of top 5 cities

The compliance rate among different agencies is also plotted, where the Buildings, Safety Engineering, and Env department has more number of tickets issued.

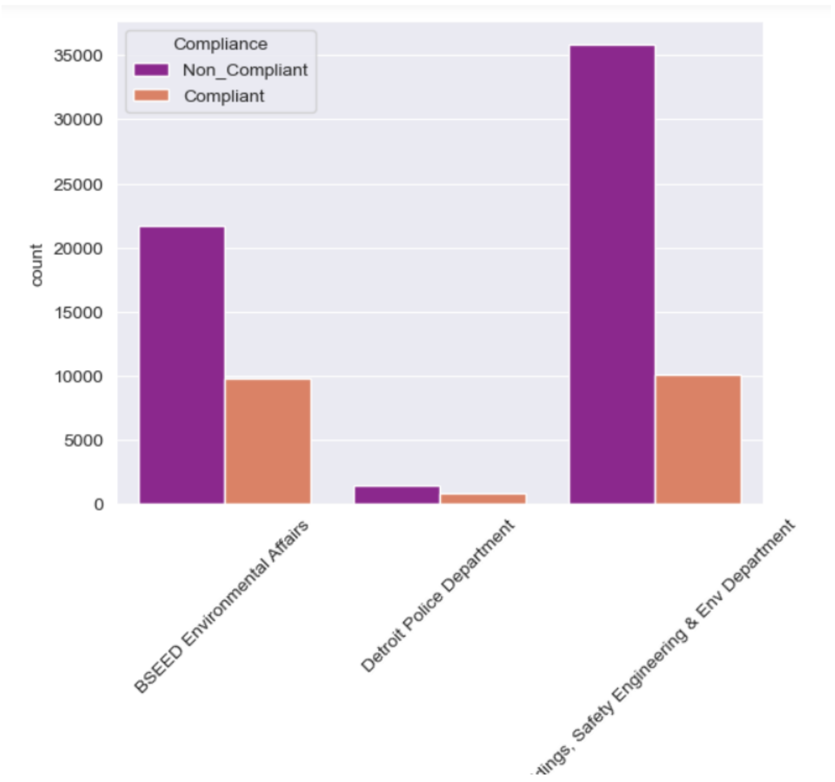
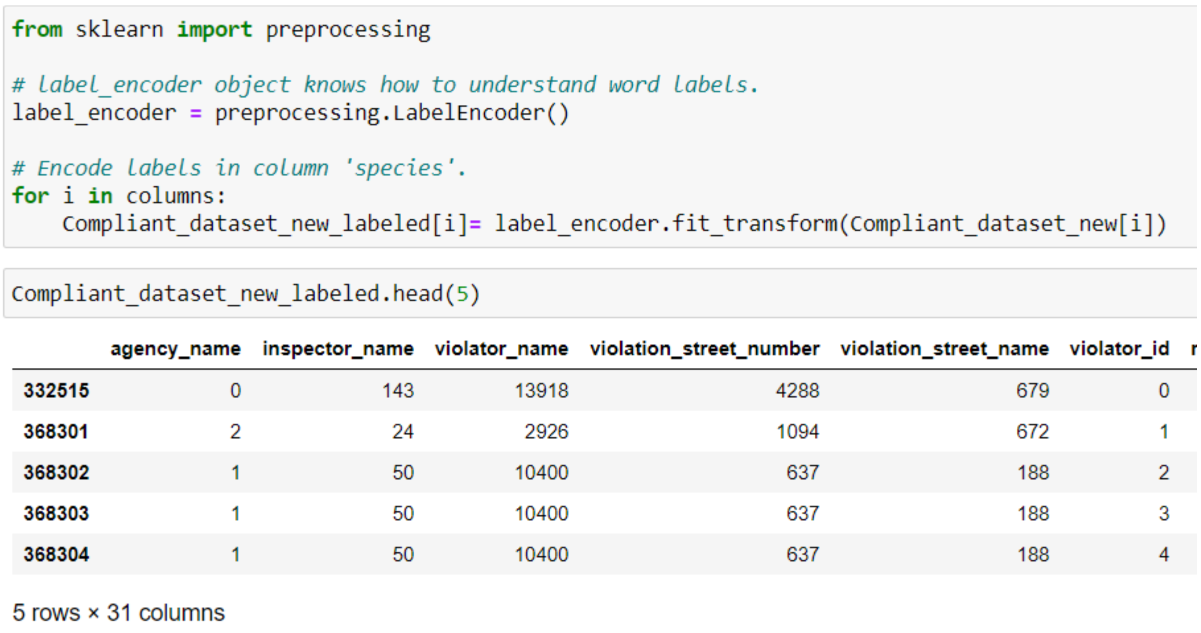


Figure 10: Compliance based on Agency

5. Feature Importance

Label encoding technique was used to convert textual information to numerical values.



```
from sklearn import preprocessing

# label_encoder object knows how to understand word labels.
label_encoder = preprocessing.LabelEncoder()

# Encode labels in column 'species'.
for i in columns:
    Compliant_dataset_new_labeled[i]= label_encoder.fit_transform(Compliant_dataset_new[i])
```

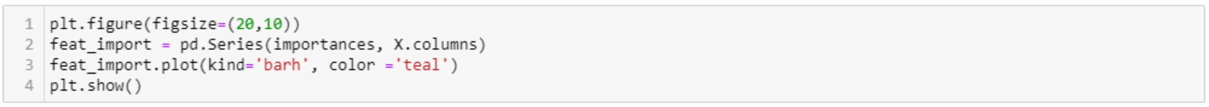
```
Compliant_dataset_new_labeled.head(5)
```

	agency_name	inspector_name	violator_name	violation_street_number	violation_street_name	violator_id
332515	0	143	13918	4288	679	0
368301	2	24	2926	1094	672	1
368302	1	50	10400	637	188	2
368303	1	50	10400	637	188	3
368304	1	50	10400	637	188	4

5 rows × 31 columns

Figure 11: Label encoding

To understand the mutual information each features contributes towards the label, three classifiers were used, which are AdaBoostClassifier, mutual_info_classif, and mutual_info_regression.



```
plt.figure(figsize=(20,10))
feat_import = pd.Series(importances, X.columns)
feat_import.plot(kind='barh', color ='teal')
plt.show()
```

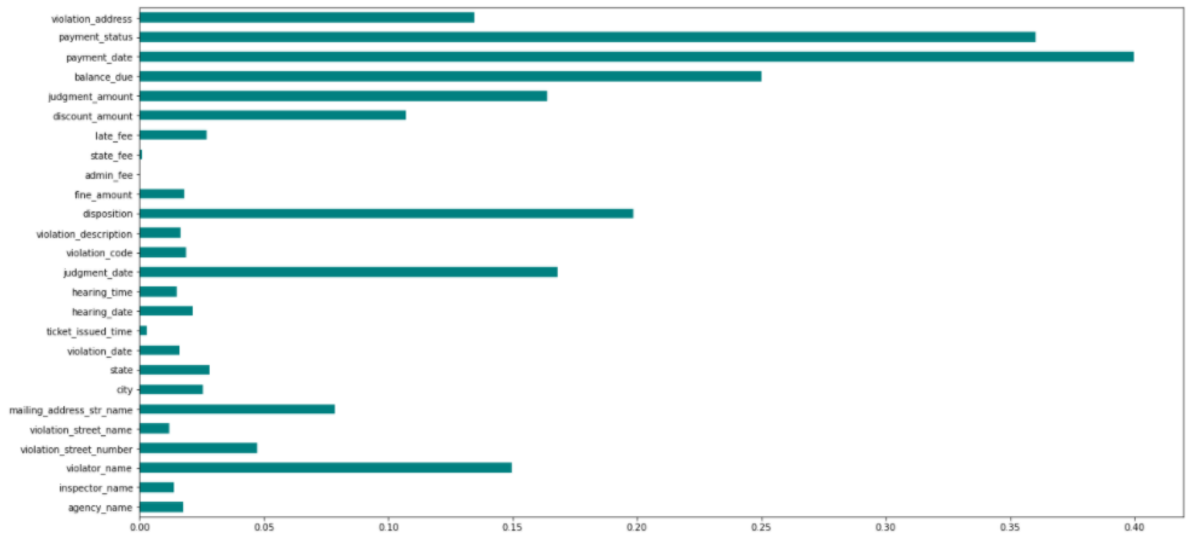


Figure 12: Mutual class info (Mutual Information)

6. Feature Selection

The selected features were chosen based upon the results drawn from the classifiers.

7. ML Modelling

There were four different machine learning models used for prediction namely Logistic Regression, Naïve Bayes, Support Vector Classification and Gradient Boosting. All the models were trained and tested with 90:10 ratio. Furthermore, k fold cross-validation was performed to make sure that models weren't overfitting.

8. Results and Conclusion

The accuracy scores were noted as following:

Model Used	**Accuracy Score**	**Average k fold score (mean)**
Logistic Regression	97.69 %	97.6 %
Naïve Bayes	91.73 %	91.8 %
SVM	97.81 %	97.66 %
Gradient Boosting	97.86 %	97.3 %

Since this is an imbalanced classification problem with non-compliant cases outnumbering the compliant ones, accuracy is not a good evaluation measure for our classification.

Therefore ROC curves were plotted to evaluate and compare the performance of four models. When interpreting probabilistic predictions, plots from the curves can be generated and used to explain the different threshold values' trade-off results.

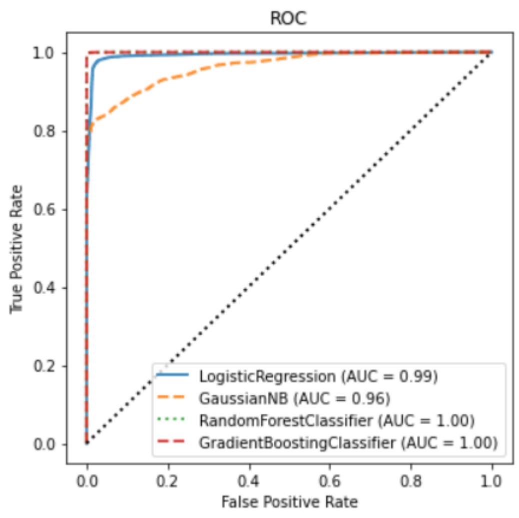


Figure 13: AUC- ROC Curve

To conclude, the following key aspects were derived:

1. For this dataset, Support Vector Classification is the model the team would suggest.

2. If a specific discount is provided in fine, the chances of a person being compliant is very high

3. If a person is non-compliant, the maximum timeline to expect for him/her to pay is 26 months, after which he can be listed as a permanent defaulter

4. Most non-compliance occurs in high fines

9. Reference

- Cell, A., Rauh, A., Tan, X., Webb, J., Bochu, J., Stroud, J., & Tan, C. (2017, September 28). Understanding Blight Ticket Compliance in Detroit. Data Science for Social Good Conference. http://www.dssgfellowship.org/wp-content/uploads/2017/09/cell.pdf
- City of Detroit Open Data Portal (n.d.). Blight Violations. https://data.detroitmi.gov/datasets/blight-violations?geometry=-83.535%2C42.259%2C-82.668%2C42.436
- Brownlee, Jason. (2020, January 6). ROC Curves and Precision-Recall Curves for Imbalanced Classification. Machine Learning Mastery. https://machinelearningmastery.com/roc-curves-and-precision-recall-curves-for-imbalanced-classification/