Crime Detection in Dallas City

Prabhleen Kaur Saini, Ujwal Shah
University of Texas at Dallas
pxs210087@utdallas.edu, uss220000@utdallas.edu

Abstract

The Spider-Sense, an extrasensory awareness of impending danger felt by Spider-Man, which helps him fight crime. Considering the alarming increase in crimes over the past few years, the law enforcement officers can definitely benefit from Spider-Sense (Rotaru, Huang, Li, Evans, & Chattopadhyay, 2022). This study introduces a novel approach to crime prediction, which could prove to be valuable to law enforcement agencies in preventing and reducing crime in their jurisdiction. To make this possible, this project proposes using machine learning techniques and massive amount of publicly available police incident data to predict areas prone to severe crime at specific days and times(Shah, Bhagat, & Shah, 2021). This project could assist police officers in route designing, resource allocation, and detecting and preventing crimes at a more accurate and faster rate.

Introduction

We utilized The Police Incident dataset, which is publicly available and provided by the Dallas Police Department (Department, 2023). The dataset encompasses information on incidents starting from June 1, 2014, with new incidents added daily. For this project, we considered data up to March 31, 2023, and the dataset contains more than one million data points with 86 features.

The first step of the project was to clean the dataset, which involved selecting relevant features based on prior knowledge, removing null values, and encoding features. Due to the enormous number of columns in the dataset containing irrelevant or redundant data, it was challenging to identify the most relevant features. Therefore, we initially considered a subset of features and gradually increased the number of features to study their impact on the training results. Initially, we considered only 7 features: Watch, Special Report, Year, Division, Month, Day, Time.

The city of Dallas is divided into seven divisions, and each division is further divided into sectors and beats (*Maps and Information*). After pre-processing, we trained the model to predict severe crimes in two steps: first over divisions and then sectors. In the first step, we trained the model to predict severe crimes based on division, which is the highest level of geographic granularity in the dataset. We analyzed the impact of various features on the training results, considering the following features.

In the second step, we employed the best model and features obtained in step one to train the model to predict severe crimes on a given day and time for sectors, which are subdivisions of divisions. As the dataset contains 37 unique sector values, this step was more challenging than the previous step. In the end we speculate our results, choice of algorithm and feature co-relation.

Data Pre-processing

The dataset had 100,000 null values that were removed. Additionally, 13 date and time columns and irrelevant information regarding the officer on duty were dropped. Multiple columns describing the victim's age, sex, and race were also removed. NIBRS Crime, NIBRS Crime Category, Crime Against, Code, and Type were also dropped as they were not related to the problem statement. Furthermore, columns like Incident Number w/year, Service Number ID, Apartment Number, RMS Code, Criminal Justice Information Service Code, and UCR Offense Description were dropped for simplicity.

Description of 15 features:

1. Watch: Police watch 1st, 2nd, or 3rd (1st watch = Late Night, 2nd watch = Days and 3rd watch = Evenings).
2. Type Location: Location type where the incident took place, for example, Apartment Parking, Residence.
3. Type of Property: The target item, such as Parking lot, Motor Vehicle.
4. Division: Geographic area comprised of census blocks where the incident occurred (smallest police geography).
5. Sector: Geographic area comprised of Sectors where the incident occurred.
6. Target Area Action Grids: Geographic areas targeted for higher-than-average crime.
7. NIBRS Group: There are two categories of offenses reported in the NIBRS: Group A and Group B. Group A offense categories make up 46

offenses. 'B' and 'C' are replaced with 0, and 'A' is replaced with 1 to create a binary classification problem (Group “A” offenses - Bureau of Justice Statistics)

8. Year: The year of the incident.
9. Month: The month of the incident.
10. Day: The day of the incident.
11. Time: The time of the incident.
12. watch2: A new column that assigns values of 1, 2, or 3 depending on the time of the report.
13. Latitude: Latitude of the location of the incident.
14. Longitude: Longitude of the location of the incident.
15. Location: The location of the incident.

Preprocessing steps include encoding values as the majority of the features were plain text. First, the y-label column 'NIBRS Group' is encoded such that 'B' and 'C' are replaced with 0, and 'A' is replaced with 1. This creates a binary classification problem.

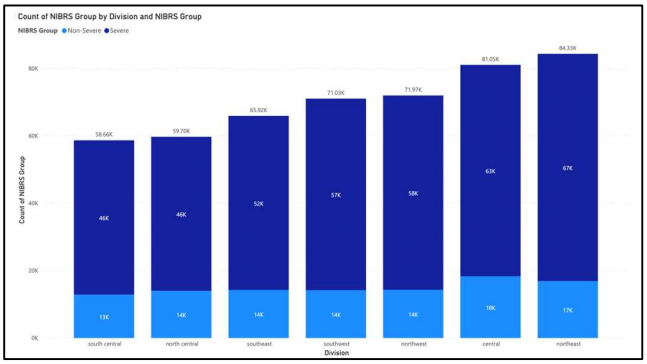
Next, the 'Date of Report' column is converted to a datetime object and then split into 'Year', 'Month', 'Day', and 'Time' columns for further analysis. The 'Time' column is later used to create a new column 'watch2' that assigns values of 1, 2, or 3 depending on the time of the report.

The 'Division' column is then encoded using LabelEncoder from sklearn.preprocessing, which assigns numerical values to each unique value of the column. This is useful for categorical variables that have no natural ordering.

Lastly, the 'Type of Property' column is encoded such that all null values are replaced with 0 and all non-null values are replaced with 1. The column is then transformed using LabelEncoder to assign numerical values to each unique value. Columns with a lot of null values like Type Location , Family Offense , were also encoded similarly.

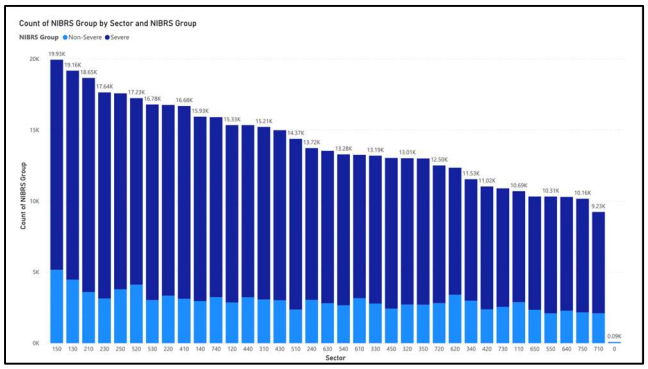
Exploratory Data Analyses

From the following graph we can observe the number of crimes were not biased towards a particular division.



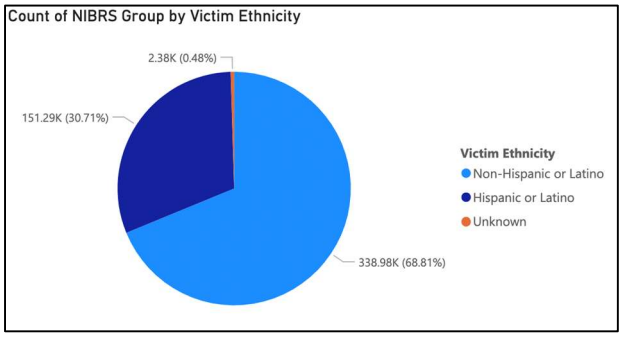
Distribution of severe and non – severe crimes over 7 divisions of Dallas City

We can observe that though some sectors are more prone to crime, sectors alone cannot be used to predict crime.



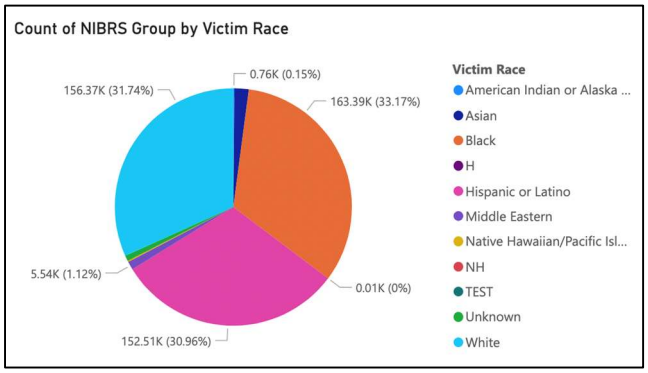
Distribution of severe and non–severe crimes over sector of Dallas City

We can observe that in 68% of reported incidents ethnicity of the victim is non-Hispanic or non-Latino



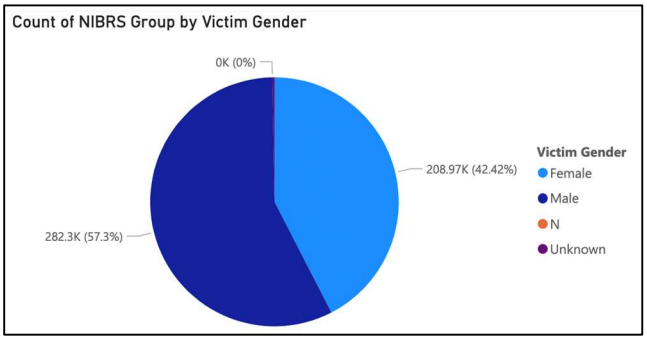
Pie Chart showing Percentage of crime on different Ethnic Groups

We can observe the maximum crimes were against people of Black race about 33%, closely followed by white and Hispanic, which is in line with the population distribution of Dallas city.



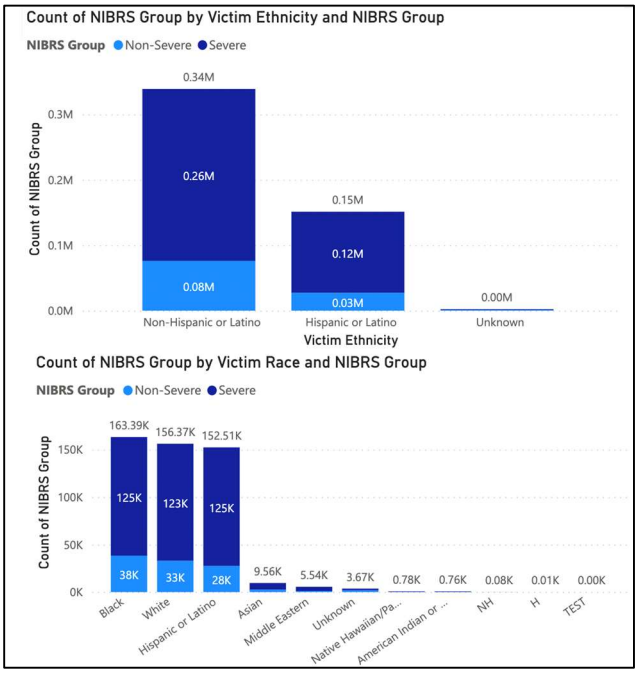
Pie Chart for distribution of crime over Race

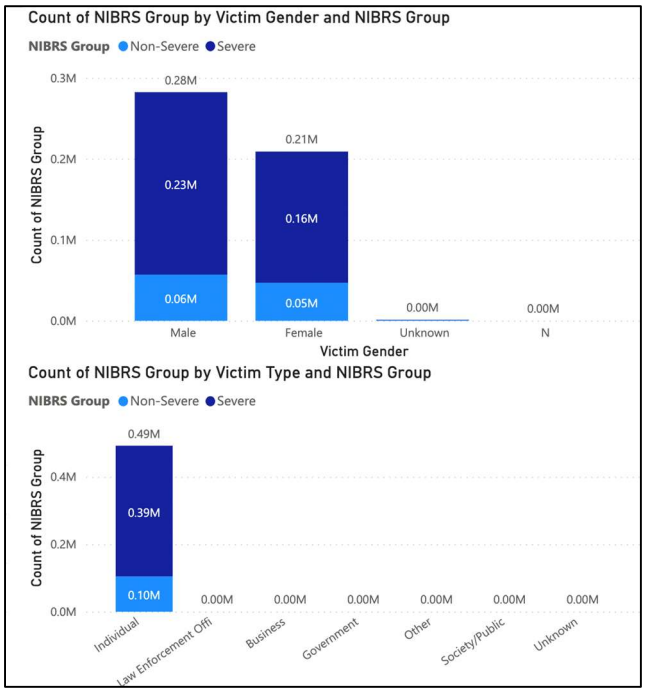
We had same proportion of males and females in the dataset, wherein males are more prone to crimes compared to women. One main reason can be that this dataset does not contain sexually oriented offenses.



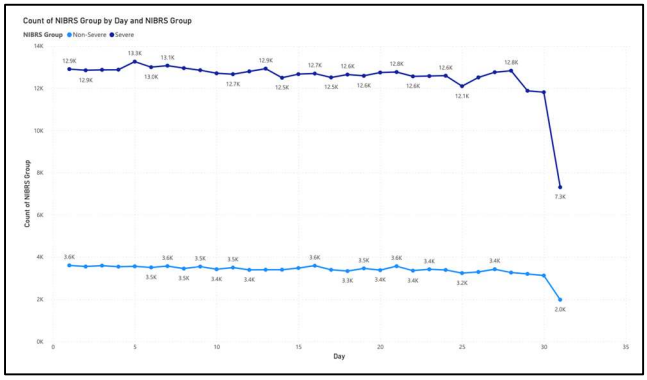
Pie Chart for distribution of crime over Gender

The following bar graphs represent distribution of severe and non-severe crimes based on victim Ethnicity, race, Gender. The dataset primarily considers offenses against individuals.



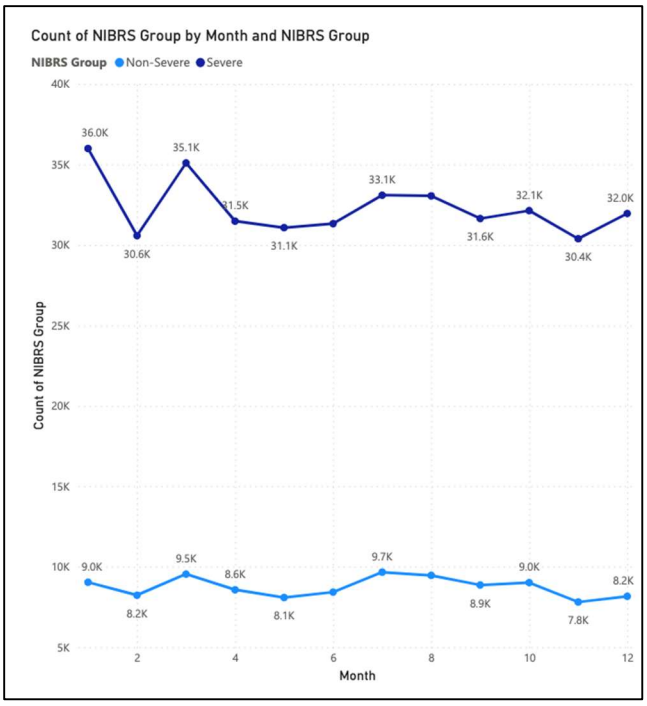


In the following graph we can observe Maximum probability of severe crime is on 5th to 7th day of a month.



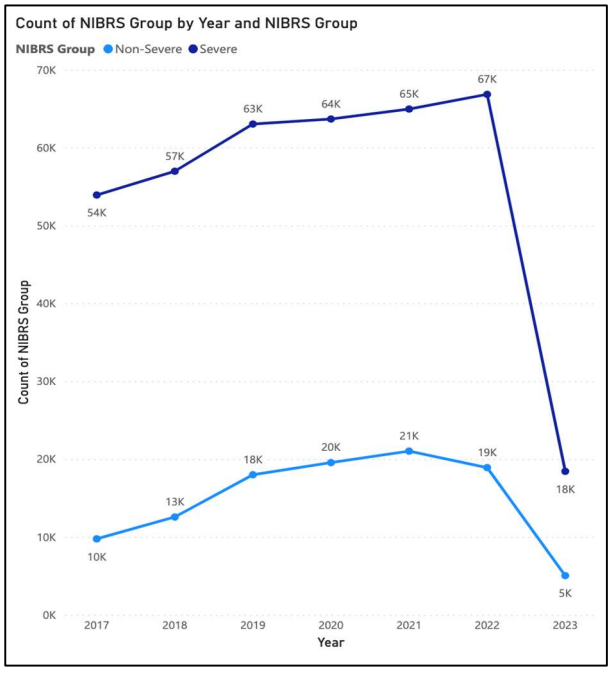
The following line graph represents the number of severe and non – severe crime-based on a particular day of month.

We can observe Maximum probability of severe crime is in the month of January closed followed by March.



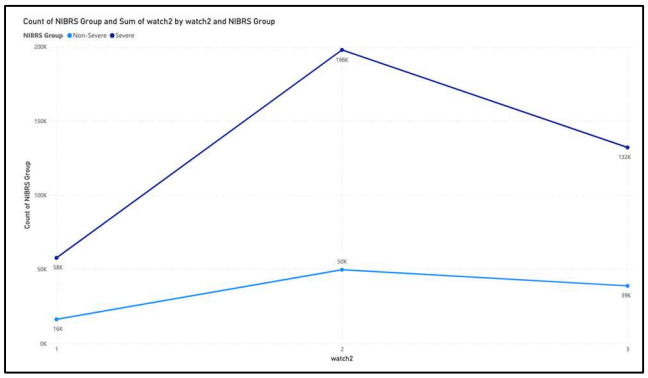
The following line graph represents the number of severe and non – severe crime-based on a particular month of the year.

We can observe an increasing trend in severe crime over the past 5 years. As we are only considering the first 3 months of 2023, there is a sharp drop in the graph.



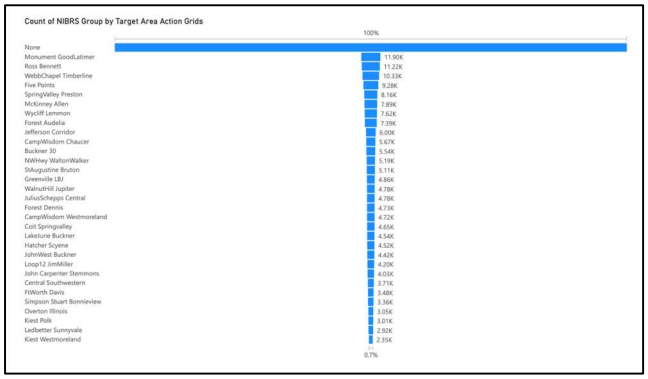
The following line graph represents the number of severe and non – severe crimes based on a particular year.

The safest time of day is from midnight 12 am to morning 7 am and the most unsafe time is morning 7 am to afternoon 4 pm.



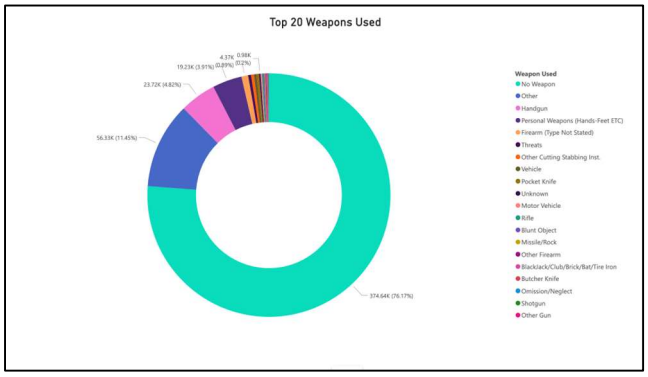
Following line graph represents number of severe and non – severe crime based on a particular time of day

We can observe that a lot of crimes happen outside the neighborhoods that were marked as dangerous by police.



Funnel chart shows distribution of crimes over target action grid which are the neighborhoods with high probability of crime, identified by the Dallas police

Handguns were the most used firearms (shotgun, rifle) for incidents in which guns were used.



Pie chart represents distribution of crimes over weapon used to commit a crime

Data Modelling

Data Collection: The data used in this study was collected from a database of criminal incidents published by the City of Dallas. The dataset was pre-processed and filtered to include only relevant features and target variables for crime detection.

Data Splitting: The pre-processed data was split into three sets - training, validation, and test sets using the train_test_split function from the Scikit-learn library. The test size was set to 0.3 to split the data into 70% for training and 30% for testing. The test set was further split into validation and test sets using the same function with a test size of 0.5.

Data Type	Size	% of Total
Train Size	344859	70
Val Size	73898	15
Test Size	73899	15
Total Dataset	492656	100

Data Pre-processing: The 'NIBRS Group' (*NIBRS* 2018) column was removed from all three sets as it was the target variable for crime detection. The indices of the three sets were also reset to ensure that they start from 0.

Model Selection: Several classification models were trained and evaluated on the pre-processed data to determine their performance in detecting crime. The models used in this study were off the shelf scikit learn models (Pedregosa et al., 1970) K-Nearest Neighbors, Decision Tree, Random Forest, Neural Network, AdaBoost, Gradient Boosting, Naive Bayes, Quadratic Discriminant Analysis, and XGBoost.

Model Training: Each of the models was trained using the respective classifier. The models' training accuracy was calculated using the score function from Scikit-learn's classifier. The validation accuracy was also calculated and stored in a dictionary with the corresponding model name. The training and validation accuracies were plotted using a bar chart to compare the models'

Model Evaluation:

Case 01 - For 800K values for 7 Divisions using 7 input features.

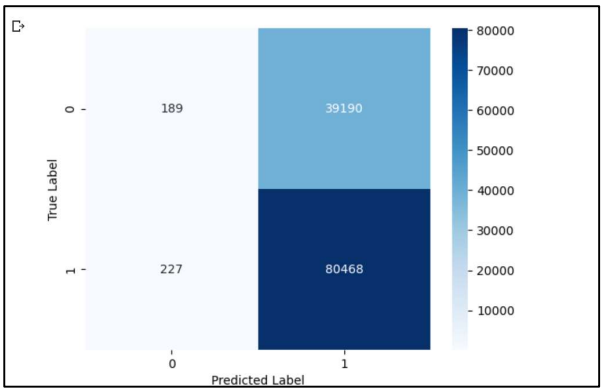
Training the model to predict severe crimes based on divisions instead of beats with all the available data resulted in simpler problems that gave the following results as shown in a figure.

Gaussian Naïve Bayes Model

Classification Report

```
Classification Report:
              precision    recall  f1-score   support

           0       0.45      0.00      0.01     39379
           1       0.67      1.00      0.80     80695

    accuracy                           0.67    120074
   macro avg       0.56      0.50      0.41    120074
weighted avg       0.60      0.67      0.54    120074
```

Confusion matrix

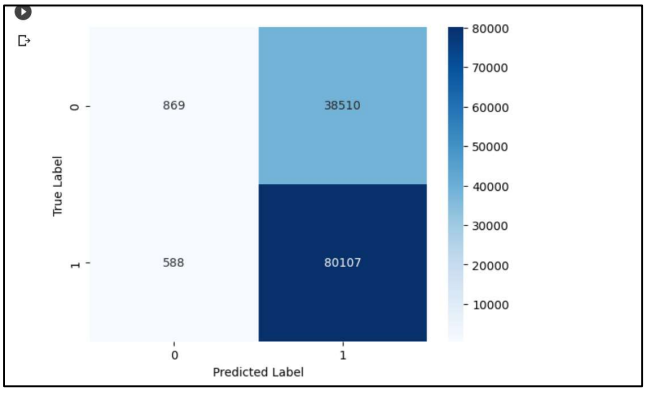


Gradient Boosting Model

Classification Report

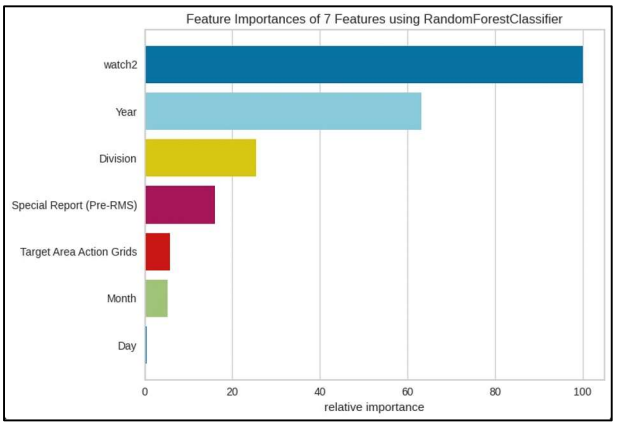
```
Classification Report:
              precision    recall  f1-score   support

           0       0.60      0.02      0.04     39379
           1       0.68      0.99      0.80     80695

    accuracy                           0.67    120074
   macro avg       0.64      0.51      0.42    120074
weighted avg       0.65      0.67      0.55    120074
```

Confusion matrix



The initial idea of predicting crime given a particular time and date proved too difficult as the evaluation metric values were not good. So, there was a need to add more input features.

The feature importance graph shows that the time of day (Watch) proved to be the most important input feature



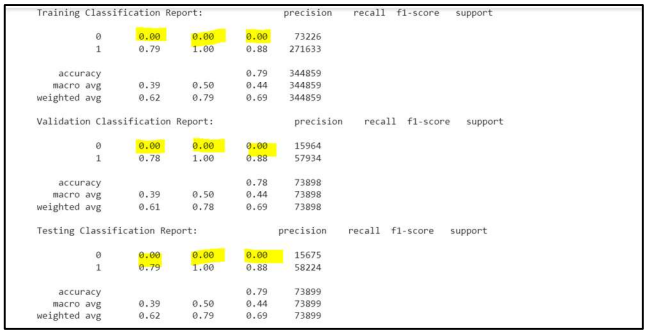
Case 02 - For 800K Dataset: based on all the 7 Divisions and 15 input features

Increasing the number of relevant features used in the model training process from seven to 15 led to a significant improvement in accuracy. Among the nine models evaluated, the Gradient Boosting algorithm produced the best results with a f1-score of 0.9948 (train), 0.8696 (validation), 0.8703 (test).

Logistic Regression

To check if the given problem linearly separable we first tried to use simple model like logistic regression

Classification Report



Training Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	73226
1	0.79	1.00	0.88	271633
accuracy			0.79	344859
macro avg	0.39	0.50	0.44	344859
weighted avg	0.62	0.79	0.69	344859

Validation Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	15964
1	0.78	1.00	0.88	57934
accuracy			0.78	73898
macro avg	0.39	0.50	0.44	73898
weighted avg	0.61	0.78	0.69	73898

Testing Classification Report:

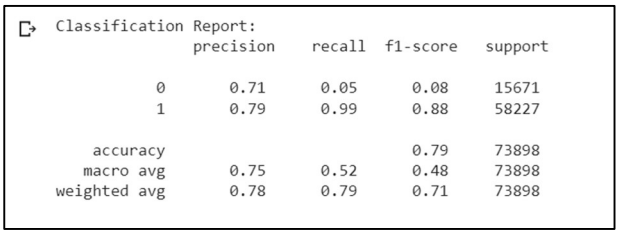
	precision	recall	f1-score	support
0	0.00	0.00	0.00	15675
1	0.79	1.00	0.88	58224
accuracy			0.79	73899
macro avg	0.39	0.50	0.44	73899
weighted avg	0.62	0.79	0.69	73899

We get 100% recall, 0 false negative which means the model fails to predict non-severe crimes, which might be due to class imbalance.

As ensemble methods are robust to class imbalance, we tried bagging and booting algorithms.

Gradient Boosting

Classification Report

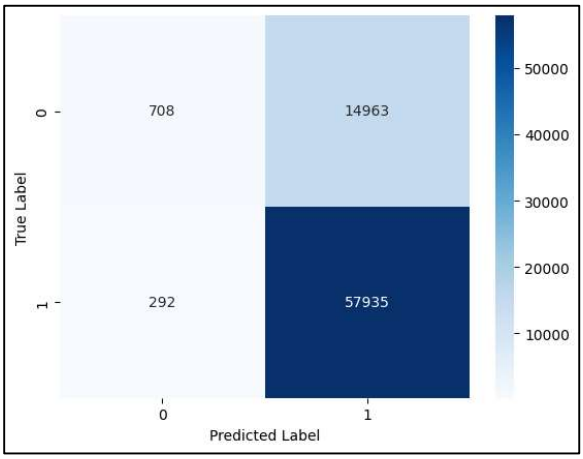


Classification Report:

	precision	recall	f1-score	support
0	0.71	0.05	0.08	15671
1	0.79	0.99	0.88	58227
accuracy			0.79	73898
macro avg	0.75	0.52	0.48	73898
weighted avg	0.78	0.79	0.71	73898

We got test f1-score 0.8815, test precision 0.7959 and test recall 0.9960.

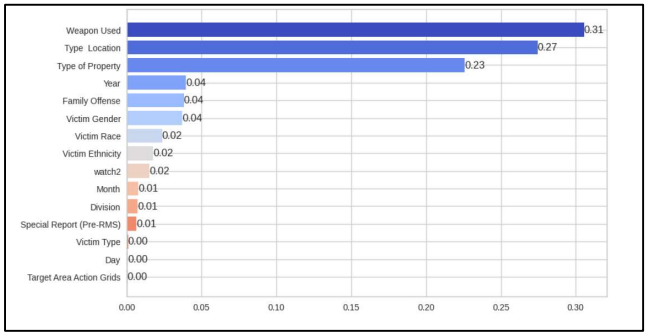
Confusion matrix



Recall is still close to one.

Contrary to watch time as the most important feature earlier, when we added more input features its effect on y-label dropped significantly.

Now, weapon used, type of location and type of property proved to be the most important features



Feature Importance of 15 input features

Case 3 - For 800K Dataset: based on all the 37 Sector and 15 input features.

The model was also evaluated for predicting severe crimes in sectors, which are subdivisions of divisions, and achieved similar results. With 37 unique values for sectors in the dataset, the best models were Bagging with train, validation, and test f1 scores as 0.99, 0.87 and 0.87.

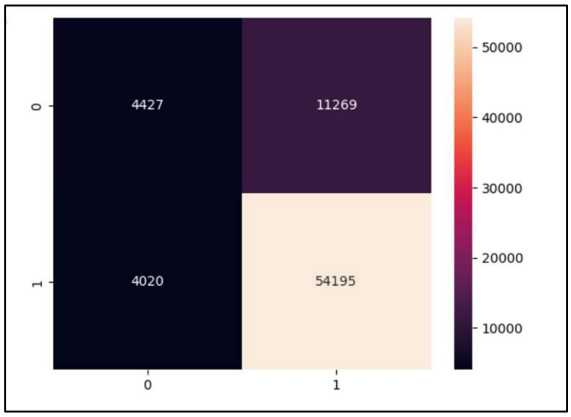
Bagging

Classification Report

```
              0       1.00      0.99      0.99     73371
              1       1.00      1.00      1.00    271547

       accuracy                           1.00    344918
      macro avg       1.00      0.99      1.00    344918
   weighted avg       1.00      1.00      1.00    344918


Validation Classification Report:
              precision    recall  f1-score   support

           0       0.53      0.28      0.37     15696
           1       0.83      0.93      0.88     58215

    accuracy                           0.79     73911
   macro avg       0.68      0.61      0.62     73911
weighted avg       0.76      0.79      0.77     73911
```

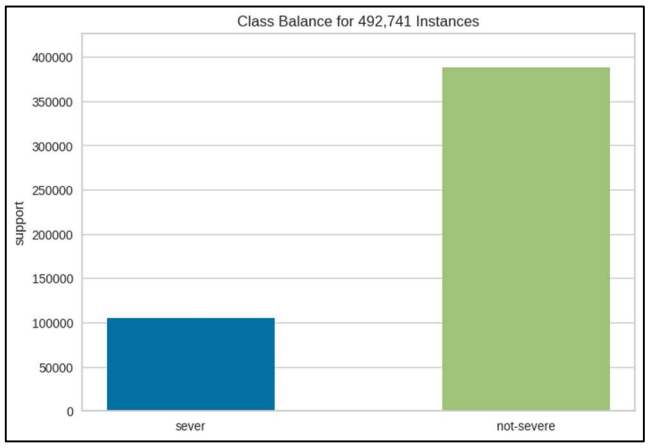
We got test f1-score 0.87, test precision 0.82 and test recall 0.92.

Confusion matrix



We got the lowest recall using bagging.

Results



There is class imbalance in the dataset hence, accuracy is a bad metric. We will use f1-score, precision and recall evaluating and compare our model's performance.

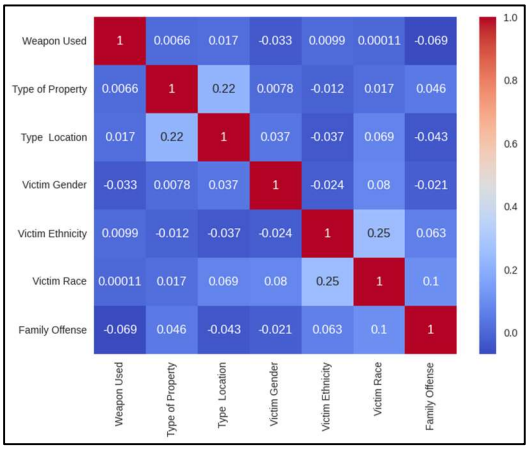
Comparative study of model performance:

Case 01: Divisions

Model	Train Accuracy	Val Accuracy	Test Accuracy	Train Precison	Val Precision	Test Precision	Train Recall	Val Recall	Test Recall	Train F1	Val F1	Test F1
Nearest Neighbors	0.8550	0.7450										
Decision Tree	0.7890	0.7895	0.7919	0.7897	0.7902	0.7924	0.9970	0.9975	0.9977	0.8822	0.8821	0.8800
Random Forest	0.7880	0.7882	0.7897	0.7868	0.7872	0.7896	0.9990	1.0000	0.9999	0.8815	0.8817	0.8793
Neural Net	0.7870	0.7881										
AdaBoost	0.7900	0.7900	0.7928	0.7911	0.7917	0.7940	0.9957	0.9957	0.9959	0.8823	0.8823	0.8803
Gradient Boosting	0.7930	0.7930	0.7952	0.7933	0.7936	0.7959	0.9959	0.9955	0.9960	0.8836	0.8837	0.8815
Naive Bayes	0.7847	0.7850										
QDA	0.7833	0.7821										
XGBoost	0.7873	0.7875	0.7895	0.7867	0.7870	0.7895	1.0000	1.0000	1.0000	0.8812	0.8813	0.8790
Bagging	0.9918	0.7829	0.7846	0.9925	0.8256	0.8241	0.9971	0.9184	0.9221	0.9948	0.8696	0.8703

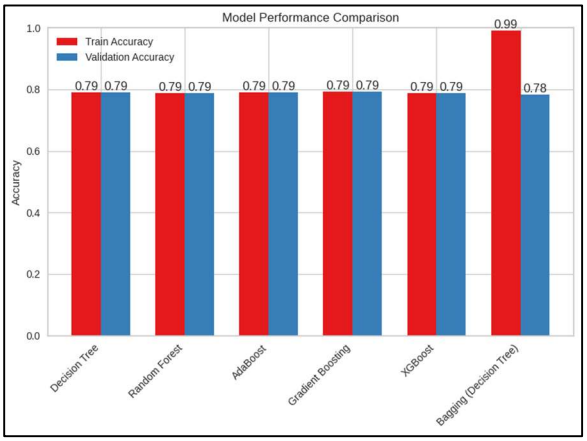
The table shows training, validation and test accuracy, precision, f1-score and recall and recall values for 9 models. Best train f1-score value was observed gradient boosting, best test recall and test precision values were observed for bagging (0.92) and (0.82)

Feature Co-relation



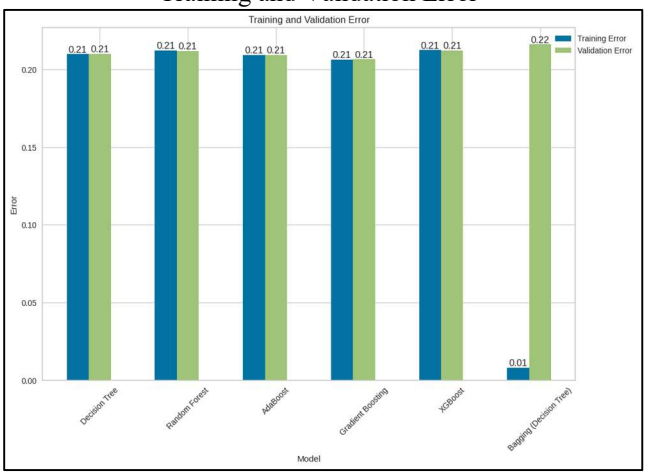
This heatmap shows that there is no strong co-relation between the newly added input features. Therefore, we can include them.

Accuracy



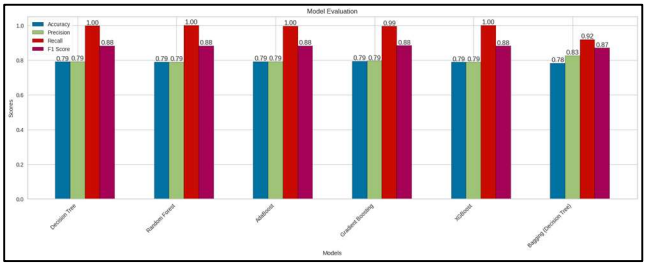
Accuracy is not a good metric as visible in the above graph, as does not help to discriminate model performances

Training and Validation Error



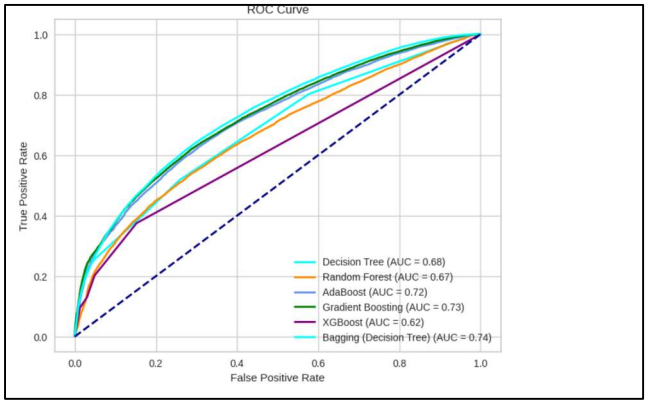
Bagging performed remarkably well in training with the lowest training error.

Precision, Recall and F1 score



A comparative study over top 5 models

ROC Curve



Bagging gave the best AUC value equal to 0.74

Case 02: Sectors

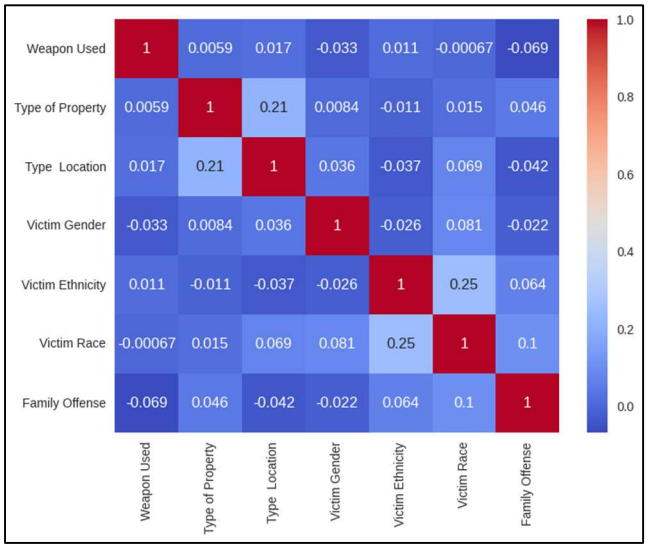
Model	Train Accuracy	Val Accuracy	Test Accuracy	Train Precison	Val Precision	Test Precision	Train Recall	Val Recall	Test Recall	Train F1	Val F1	Test F1
Nearest Neighbors	0.8552	0.7454										
Decision Tree	0.7901	0.7878	0.7926	0.7892	0.7911	0.7934	0.9975	0.9975	0.9974	0.8819	0.8824	0.8806
Random Forest	0.7875	0.7861	0.7912	0.7865	0.7887	0.7911	0.9999	0.9998	1.0000	0.8810	0.8813	0.8797
Neural Net	0.7875	0.7861										
AdaBoost	0.7909	0.7888	0.7932	0.7909	0.7928	0.7951	0.9954	0.9951	0.9949	0.8820	0.8827	0.8810
Gradient Boosting	0.7935	0.7913	0.7961	0.7928	0.7950	0.7972	0.9954	0.9956	0.9953	0.8834	0.8839	0.8825
Naive Bayes	0.7868	0.7855										
QDA	0.7867	0.7855										
XGBoost	0.7874	0.7861	0.7905	0.7860	0.7882	0.7905	1.0000	1.0000	1.0000	0.8810	0.8812	0.8797
Bagging	0.9968	0.7922	0.7889	0.9960	0.8274	0.8243	0.9990	0.9302	0.9292	0.9970	0.8757	0.8736

The table shows training, validation and test accuracy, precision, f1-score and recall and recall values for 9 models.

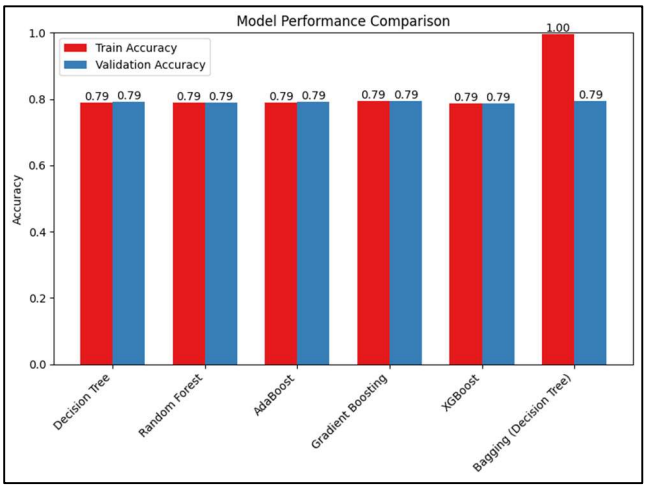
Best train f1-score value was observed gradient boosting, best test recall and test precision values were observed for bagging (0.92) and (0.82)

The following figures show similar results to those observed over division.

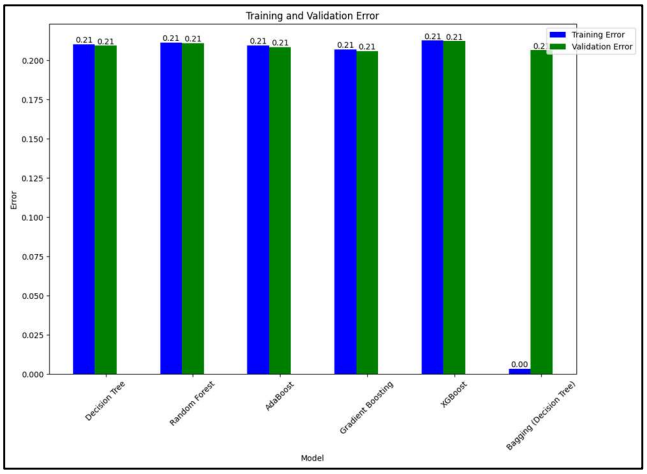
Feature Co-relation



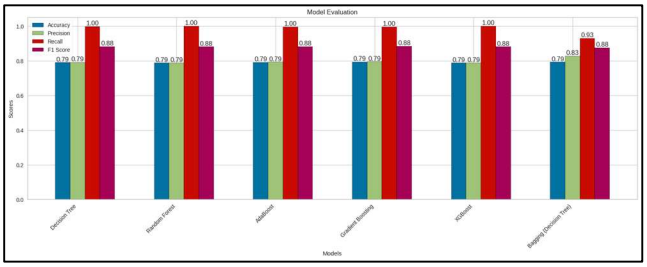
Accuracy



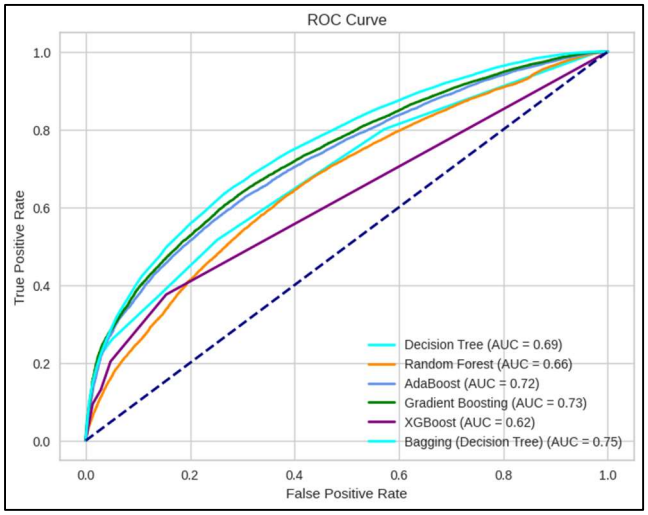
Error



Precision, Recall and F1 score



ROC Curve



Best model – Bagging (n_estimator = 100 & base_estimator=Decision Trees).

Best Evaluation metrics: F1-Score, or ROC_AUC Curve.

Discussions

The initial model was trained using a minimum number of input features, mainly using date and time to predict crime, which led to inaccurate results. Although we achieved an accuracy of 67%, the recall was 1, which meant the model always classified all crimes as severe. Moreover, there was no single division that had a majority of crimes, proving that the problem was not linearly separable. The low performance of a simple model like logistic regression confirmed that the problem was not solvable using simple models.

Significant improvements were not observed by using a complex model like gradient boosting. The primary reason for the low accuracy of this model was the small number of input features. During the initial analysis, we discarded many relevant features that had several null values, such as "weapons used." But then, we encoded the null values as 0, assuming that in case of null values, no weapons were used. Similarly, we handled other features like type of property & for other variables like Type of Location,etc. Our assumption was correct. Some of the newly added features like "Weapon Used" were among the most critical features, which made sense as crimes involving weapons tend to be more severe than those that do not. We observed that

accuracy was not a good metric due to class imbalance in the dataset; hence we used f1-score to choose the best model.

We were able to replicate our results in sectors that had more unique values than divisions, and the feature importance and correlation remained similar. This proved that our approach could be successfully replicated for smaller geographical regions too.

We believe that by hyperparameter tuning, we can reproduce comparable results for the subdivision of sector-beats. Even in its current form, the model is useful for law enforcement officers to deploy more task forces in sectors with more severe crimes during a given watch (patrolling time).

References

Crime analytics dashboard. (n.d.). Retrieved May 7, 2023, from https://www.dallasopendata.com/stories/s/r6fp-tbph

Department, D. (2023, May 02). Police incidents: Dallas opendata. Retrieved May 7, 2023, from https://www.dallasopendata.com/Public-Safety/Police-Incidents/qv6i-rri7

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, É. (1970). Retrieved from https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

Group “A” offenses - Bureau of Justice Statistics. (n.d.). Retrieved May 8, 2023, from https://bjs.ojp.gov/sites/g/files/xyckuh236/files/media/document/offensea_offenseb.pdf

(N.d.). Retrieved from https://dallaspolice.net/division/southeast/mapsandinformation

NIBRS. (2018, September 10). Retrieved May 7, 2023, from https://www.fbi.gov/how-we-can-help-you/more-fbi-services-and-information/ucr/nibrs

Shah, N., Bhagat, N., & Shah, M. (2021). Crime forecasting: A machine learning and computer vision approach to crime prediction and prevention. Visual Computing for Industry, Biomedicine, and Art, 4(1). doi:10.1186/s42492-021-00075-z