University of Antwerp
Faculty of Business
and Economics

Data Engineering Project

Crimes in Chicago

Professor Len Feremans

Group 3

Khanh Linh Kha	20232618
Begüm Tamer	20232064
Zaineb Achoukhi	20181374
Yentl De Maere	20201134

Contents

Data Understanding

The Chicago crimes dataset shows reported crime incidents, excluding murders where data exists for each victim. The dataset is obtained from the Chicago Police Department's CLEAR system. Addresses are only shown at the block level, with specific locations withheld purposely to protect the privacy of crime victims. This dataset is continuously updated to provide timely insights into ongoing criminal activities within the city. The dataset contains numerous fields providing detailed information about each reported incident, including unique identifiers, dates, locations, crime classifications, arrest information, and geographic coordinates. Notably, the dataset includes about 7.9 million records, offering a comprehensive view of crime trends and patterns over the specified time frame ranging from 2001 to 2017.

To gain insight into the dataset, we made use of the check_df function. It starts by providing the dataframe's shape, presenting an initial understanding of the size in terms of rows and columns. Subsequently, the function shows the data types of each column, to better comprehend the values and the selection of suitable data manipulation techniques. It also offers a preview of both the beginning (head) and end (tail) of the dataframe, thus creating a visual assessment of the data's structure and layout. Lastly, we checked missing values across different columns, which helps in the understanding of data completeness and guiding subsequent data preprocessing steps.

Data Pre-processing

Pre-processing the Chicago crimes dataset involves several steps to improve the quality and usability for following analyses or modeling tasks. First, duplicate records are removed from the dataset based on the columns 'ID' and 'Case Number', to ensure data integrity and avoid redundancy. This step reduces the dataset's size to 6,170,812 data instances. Next, uninformative columns like 'Unnamed: 0', 'ID', 'Case Number', and 'Updated On' are dropped from the dataset. We consider these columns as irrelevant for analysis or modeling purposes.

Afterwards, date columns are converted to the appropriate datetime format using the pd.to_datetime function. Hereby new columns are created to store components of the date like time, date and time of day, providing further insights into time-based patterns within the data. To reduce redundant information, we suggest dropping the 'Date' column, because the same information is stored in other date-related columns.

Subsequently, missing values are addressed through different methods. For numerical columns like 'X Coordinate', 'Y Coordinate', 'Latitude', and 'Longitude', missing values are imputed with the mean value of each respective column. Categorical columns like 'Community Area', 'Ward', 'District', 'Location', and 'Location Description' are imputed with the mode value.

Finally, we considered outlier treatment, particularly for columns like 'X/Y Coordinate', 'Longitude', and 'Latitude'. However, after analyzing the distribution of the values, we noticed that the distribution of values does not show significant clustering, skewness, or irregularities that would necessitate outlier treatment. Thus, we decided that outlier treatment is not necessary for these columns.

Overall, these pre-processing steps ensure that the Chicago crimes dataset is cleansed, standardized, and prepared for further analysis tasks, enhancing the interpretability of insights derived from the data.

Exploratory Data Analysis

During the exploratory data analysis task, we concentrated on the target variable for the machine learning task: ‘Arrest’. That way, the exploratory task is complementary and can therefore support insights from the machine learning part. It could be used for example to get more insight in the feature importance graphs for the corresponding machine learning models.

Firstly, we examined if the target variable ‘Arrest’ was skewed or not. Since the target variable is a boolean variable, skewed means an imbalance in the class priors “True” and “False”. Figure 1 indicates that ‘Arrest’ is quite skewed. Specifically, 71.7% of the instances are not arrested, whereas 28.3% of the instances are arrested. Consequently, due to this class imbalance, one should not use accuracy as a metric for the evaluation of different models (Provost & Fawcett, 2013).

Which crimes occurred the most when an arrest was made?

Since we now know that the criminal got arrested 28.3% of the time, it might be interesting to know for what he or she got arrested by the police. For this analysis, the instances are grouped by ‘Primary Type’ for which ‘Arrest’ is “True”. In addition, a threshold of 75% is used for the counts. That way, only the most important primary types are displayed in Figure 2. It shows that the three most occurring crimes when an arrest was made are: narcotics, battery and theft. One notes that narcotics occur far more than battery or theft. Therefore, we looked further into this by examining which description was linked the most when a crime was categorized as ‘narcotics’.

To do so, the data is grouped on ‘Description’ and a threshold of 75% is used for the counts. Figure 3 indicates that with regard to narcotics, most people are arrested for the possession of cannabis, followed by the possession of crack and heroin.

Where did some crimes occur more or less when an arrest was made?

In the previous section, we concluded that narcotics, battery and theft are the three most occurring crimes when an arrest is made. We are now interested in which districts these crimes occur the most, relatively to others. For this analysis, we filtered the arrested incidents and grouped them by ‘District’ and ‘Primary Type’. A threshold of 75% is used for the ‘Primary Type’ counts of a corresponding district. Since there are 25 districts in the dataset, it was decided to implement only three districts in the report: district 1, district 11 and district 17 (Figures 4-6), as these districts identified different most occuring crimes. It is noted that there is more theft committed in district 1, relative to other crimes. District 11 is known for narcotics-related crimes, whereas in district 17, narcotics and battery occur almost equally.

When did the police arrest someone?

Finally, we conducted an analysis to answer the question: ‘On what time of the day are most people arrested?’. For this part, the date of each instance was firstly transformed to a '%m/%d/%Y %I:%M:%S %p'-format so that the hour can be extracted. Finally, 4 bins: [-1, 6, 12, 18, 24], with the corresponding times of the day: ["Night", "Morning", "Afternoon", "Evening"] were created. Consequently, a new feature ‘Time of Day’ was created, on which was grouped for this analysis. Figure 7 illustrates that the police arrest more criminals during the evening and afternoon, while they arrest less during the night. In the previous section, it was concluded that narcotics, battery and theft are the most occurring crimes. Therefore, we analyzed the distribution for their arrests based on time of the day. These three figures (Figures 8-10) underline that the distribution for narcotics arrests is mostly during evening and afternoon. With regard to battery, one remarks that criminals commit more during the night compared to narcotics and theft. Furthermore, theft is committed the most during the afternoon. The morning also has a fair share in this distribution.

Machine Learning

In the machine learning phase, we chose to extract a subset from the dataset to optimize computational resources given the large dataset's size. Our focus will be on building predictive models, with the target variable being “Arrest”. This variable has binary values of “True” and “False”, with “True” indicating that an arrest was made, while “False” means no arrest. Through this approach, we aim to gain valuable insights into which features and model types are effective for predicting arrests.

Data sampling

After conducting initial data preprocessing, which involved tasks similar to aforementioned preprocessing steps, such as removing duplicates, removing uninformative variables from the merged dataset, and optimizing the 'Date' feature format to '%m/%d/%Y %I:%M:%S %p', a random sampling process was carried out on the merged dataset, resulting in a selection of 100,000 samples. The decision to choose for 100,000 samples was to have a sufficiently large dataset to train machine learning models effectively and ensure manageable computational resources for processing. Although this number of samples is not fully representative of the entire dataset, which has a total of more than 6 million data points, the selection of 100,000 samples still provides valuable insights and patterns present in the dataset. Notably, we observed that the frequency distribution of the target variable in the subset is the same as that of the entire dataset. We decided that the imbalance was acceptable and that there was no need to implement any techniques to handle imbalanced data.

Training, validation and test set

Our next step involved splitting the dataset into training, validation, and test sets to avoid data leakage in the data pre-processing steps. The validation set is used for fine-tuning the model's performance, while the test set is specifically for assessing how well the model predicts real-world outcomes.

Missing values treatment

There are nine variables in the sample dataset that contain missing values. The missing values within discrete and nominal variables, including 'Community Area', 'Ward', 'Location', 'Location Description', 'District', are filled using the mode of the training set in order to avoid data leakage. For continuous data such as 'X Coordinate', 'Y Coordinate', 'Latitude', and 'Longitude', missing values are imputed with the mean of the training set.

Feature selection

Feature selection was conducted to identify the most significant features for predicting target variables before model training. This decision was made with the aim of reducing the number of variables for several reasons. Firstly, by focusing on the most relevant features, we aim to improve the performance and efficiency of our predictive models. Moreover, working with a smaller set of features can help reduce the computational requirements for training and deploying models, especially when dealing with large datasets. It will also allow us to optimize the preprocessing steps and to have clearer insights into which specific variables are most important to predictions.

To conduct feature selection, we chose to initially train a random forest model on the sample dataset with all available features. This model was trained on the training set, fine-tuned using the validation set, and evaluated for performance on the test set. The reason for choosing the random forest model at this stage is because it can handle multiple data types without the need for feature standardization or normalization during training. Additionally, for categorical features, we encoded them into numerical categories, and for the 'Date' feature, we transformed it into seconds and labeled it as 'timestamp' before feeding them into the algorithm.

Subsequently, we extracted feature importance scores to identify the most influential features (Figure 11, Table 1). To further validate our selections, we also cross-checked these results with the correlation matrix (Figure 12). We then proceeded to choose features with an importance threshold above 0.045, which include 'Primary Type', 'FBI Code', 'Description', 'IUCR', 'timestamp', 'Location', 'Block', 'Location Description'. For the next pre-processing steps, we used the original values of these features without the factorization applied earlier.

Variable binning and encoding

In the pre-processing phase, we had variables such as 'Primary Type,' 'Description,' and 'Block,' each containing numerous distinct categorical values. However, many of these categories had very low occurrence counts. To address this for each feature, inspired by Rare encoding technique, we created a new category, labeled as 'Others' to consolidate these rare instances. This method involves combining those categories having few occurrences into a single category, thereby simplifying the variable. To identify rare values, we established thresholds based on the specific occurrence rate of such values within the dataset that we observed to be fewer than the occurrences of other values. If a value's occurrence is below the threshold, it would be classified into the "Others" category.

For the 'timestamp' feature, we performed data normalization to ensure that the feature contributes equally to the similarity measure. During standardization, we exclusively considered the mean and standard deviation of the variable in the training set to ensure there is no data leakage.

As there are many types of locations recorded in the 'Location Description' feature, we decided to categorize them based on their shared characteristics. We established five categories: 'RESIDENCE', 'COMMERCIAL', 'STREET', 'COMMUNITY' and 'OTHER'. Each type of location within the feature was then mapped to one of these groups. For example, entries such as 'RESIDENCE,' 'APARTMENT,' or 'HOUSE' were grouped under the 'RESIDENCE' category. Any entries not covered by this mapping were assigned to the 'OTHER' category.

We extracted latitude and longitude values from the 'Location' column and used them to initialize the K-means algorithm (Holbrook & Cook, n.d.). Choosing the three most common coordinates, we set the number of clusters (denoted by 'K') to 3. These coordinates served as initial cluster centers. The algorithm then assigned each location to one of the three clusters based on proximity. After clustering, we updated our dataframe by adding a new 'Cluster' column, indicating each location's assigned cluster.

For nominal variables with many distinct values, specifically 'FBI Code' and 'IUCR', the Weight of Evidence (WoE) method was applied (Moeyersoms & Martens, n.d.; Packt, n.d.). While implementing WoE, the calculations derived from the training set were mapped and applied to the corresponding variables in the validation and test set to avoid data leakage.

After the binning steps, we performed dummy encoding for each feature. The processed dataset would be exported to an Excel file named "Preprocessed_data.xlsx" to facilitate further stages of modeling and evaluation.

Modeling and evaluation

After completing the data preprocessing steps, we proceeded to the application of 8 different machine learning models to optimize predictive performance, namely K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Logistic Regression, AdaBoost, CatBoost, XGBoost, and Neural Network. For neural networks, our decision to use TensorFlow was due to its scalability and performance optimizations, making it suitable for training deep learning models even on datasets with millions of rows (GeeksforGeeks, 2024).

Each model underwent a hyperparameter tuning process, aiming to identify the configurations that enhance predictive accuracy. In the process of evaluating different models, the results of accuracy and AUC scores of the test set were recorded (Table 2). We compared the predictive performance of different models based on ROC-AUC scores of the test set.

Of the 8 models mentioned, the best performer was CatBoost, with an AUC test score of 0.903 (ROC curve as illustrated in Figure 13), then came Neural Networks with a close score of 0.9029 and XGBoost with 0.9025.

Using SHAP values (SHapley Additive exPlanations) with CatBoost helped identify key predictive features like "IUCR_WoE", "FBI_Code_WoE", and "Location_Category_RESIDENCE" (Awan, 2023). We then analyzed CatBoost's learning curve, which suggested minimal overfitting, with training and cross-validation accuracies closely aligned, indicating generalization to unseen data (Figure 14).

Insights and recommendations

The Random Forest model experienced a drop in the AUC score (from 0.89 with only the factorized dataset when performing feature selection) to 0.85 after employing encoding methods. However, CatBoost and XGBoost models demonstrated consistent performance despite similar changes. The difference in feature importance between datasets for the best model, CatBoost, was observed with SHAP. With the factorized dataset, features like 'Primary Type' were seen as most important. Conversely, in the dataset with selected features, features like 'IUCR_WoE' and 'FBI_Code_WoE' were most important. The decrease in importance of features like 'Location' and 'Primary Type' post-encoding could be due to the grouping of raw categorical information into broader categories that change their impact on predictions (Figures 15-16).

Using the entire dataset aims to maximize data diversity and volume for training and testing our predictive model. However, for future practices, splitting the dataset by time intervals could provide insights into the model's adaptability to changing trends and its predictive accuracy. This analysis can enhance the model and its applicability to real-world scenarios.

In summary, the application of machine learning is useful and can be improved in various aspects. There are a number of practical implications for using a machine learning model to predict whether an individual would be arrested based on crime-related variables. Predictive models can help law enforcement organizations prioritize cases based on the likelihood of an arrest, and help prevent crime by recognizing patterns or trends that lead to arrests. They can also provide insights to the police so that they can distribute resources more effectively to respond to crimes. However, there are a number of considerations and challenges to take into account before using a machine learning model to predict arrests based on crime data. For example, making sure that the model's predictions are transparent and explainable is important because law enforcement decisions can lead to substantial consequences. Moreover, it is crucial to check the model for biases and fairness, especially related to race, gender, and other sensitive status. Since crime data often includes private information about individuals, there is a need to keep that data safe and secure when using these models. Additionally, the usage of machine learning should also follow relevant regulations such as GDPR to ensure compliance and protect individuals' rights.

Conclusion

In conclusion, this project applied data preprocessing techniques, exploratory data analysis and machine learning models to understand various crime aspects and predict arrests from the Chicago crimes dataset. Key steps included data understanding, preprocessing, exploratory analysis, feature selection, variable binning and encoding, and modeling with 8 different algorithms. Regarding machine learning, the CatBoost model achieved the best performance with an AUC score of 0.903 on the test set, demonstrating its effectiveness in predicting arrests based on crime-related variables. Feature importance analysis highlighted the significance of encoded features like 'IUCR' and 'FBI_Code' in making accurate predictions.

Future work could involve splitting the dataset by time intervals to assess the model's adaptability to changing crime trends. Continuous refinement and validation of the models are necessary to maintain their effectiveness and reliability over time. Overall, this project shows the potential of exploratory analysis and machine learning in assisting law enforcement to prioritize cases, allocate resources effectively, and prevent crime by identifying patterns that lead to arrests. However, it is crucial to address considerations such as ensuring transparency, fairness, data security, and regulatory compliance when implementing these models in real-world scenarios.

The use of ChatGPT

For our project, we've utilized ChatGPT at some stages. Specifically, we used it to refine and polish the sentences in our report, helping to improve clarity and comprehensiveness while reducing repetitiveness and grammatical errors. During the data preparation phase, ChatGPT was useful in helping us to briefly understand certain concepts and techniques related to binning and encoding strategies for different variable types. It particularly recommended the WoE and K-means binning techniques for high-cardinality nominal variables. After understanding these concepts and cross-referencing them with reliable sources, we could decide the appropriate methods for our project. While ChatGPT assisted our writing process, it's important to note that all ideas, topics, and initial drafts of text originated from team members. Additionally, there are still some drawbacks of ChatGPT during our usage, such as its tendency to generate complex words and abstract ideas. Therefore, we had to verify these concepts and ideas with other reliable sources, as well as revise and rephrase the content to make it easier to understand and avoid plagiarism.

References

Awan, A. A. (2023). *Using SHAP Values for Model Interpretability in Machine Learning*. KDnuggets. https://bit.ly/46J7oVK

Crimes in Chicago. (n.d.). [Dataset]. Kaggle. https://www.kaggle.com/datasets/currie32/crimes-in-chicago

GeeksforGeeks. (2024). *Introduction to TensorFlow*. GeeksforGeeks. https://www.geeksforgeeks.org/introduction-to-tensorflow/

Holbrook, R., & Cook, A. (n.d.). *Clustering With K-Means*. Kaggle. https://www.kaggle.com/code/ryanholbrook/clustering-with-k-means/tutorial

Moeyersoms, J., & Martens, D. (n.d.). *Data Mining Tip: How to Use High-cardinality Attributes in a Predictive Model*. KDnuggets. https://www.kdnuggets.com/2016/08/include-high-cardinality-attributes-predictive-model.html

Packt. (n.d.). *Encoding with the Weight of Evidence*. https://subscription.packtpub.com/book/data/9781804611302/2/ch02lvl1sec19/encoding-with-the-weight-of-evidence

Provost, F., & Fawcett, T. (2013). *Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking*. O’Reilly Media.

Appendix

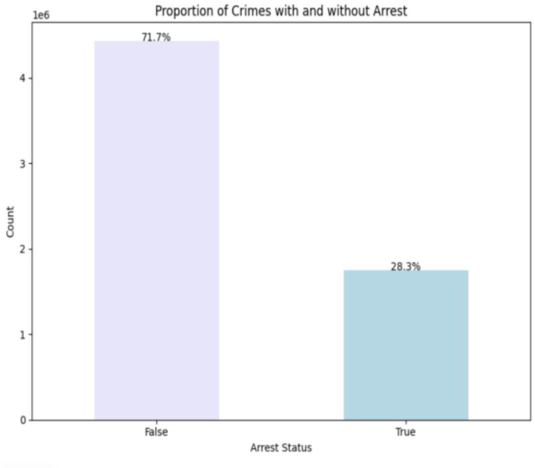

Figure 1: Class priors of the target variable 'Arrest'

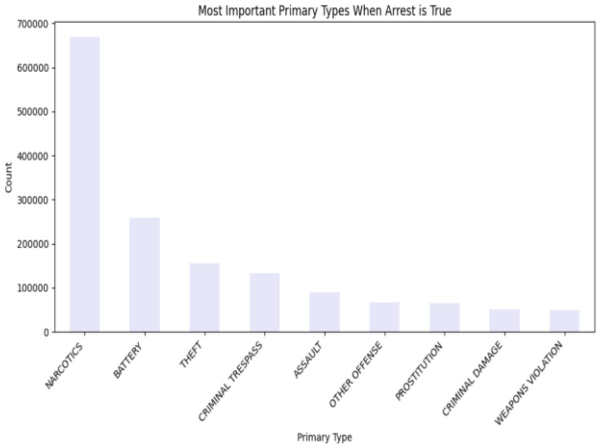

Figure 2: Most occurring crimes when arrested

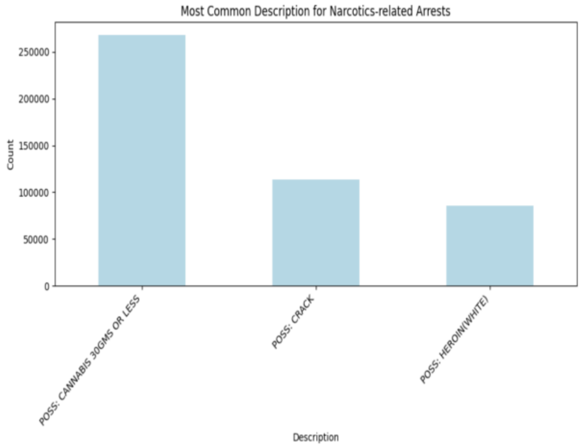

Figure 3: Most common description when 'Primary Type' is NARCOTICS

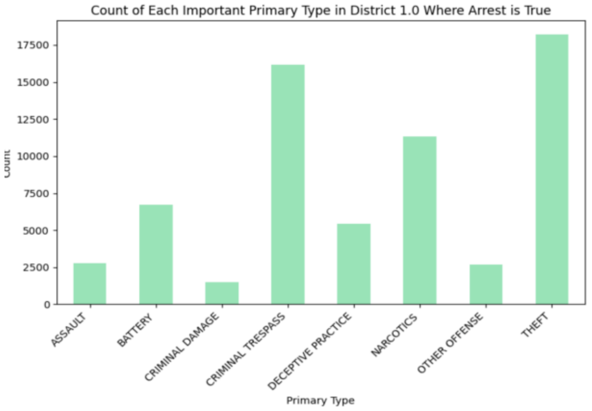

Figure 4: Prevalence of the most occurring crimes in district 1

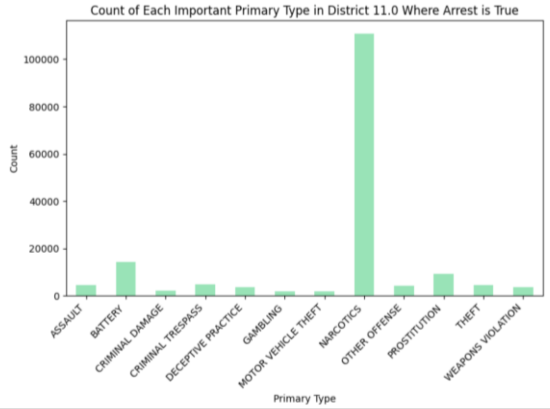

Figure 5: Prevalence of the most occurring crimes in district 11

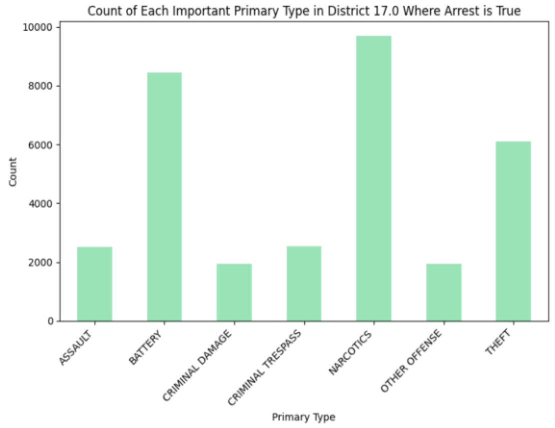

Figure 6: Prevalence of the most occurring crimes in district 17

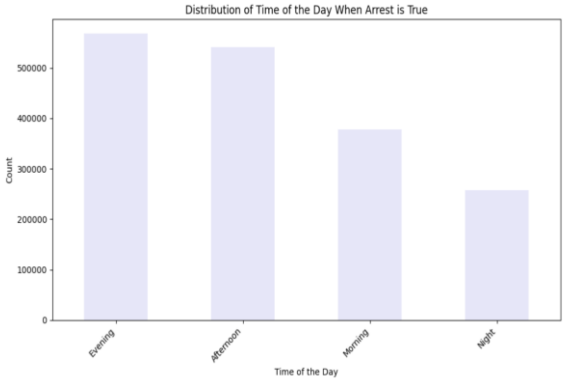

Figure 7: Distribution of arrests for the time of the day

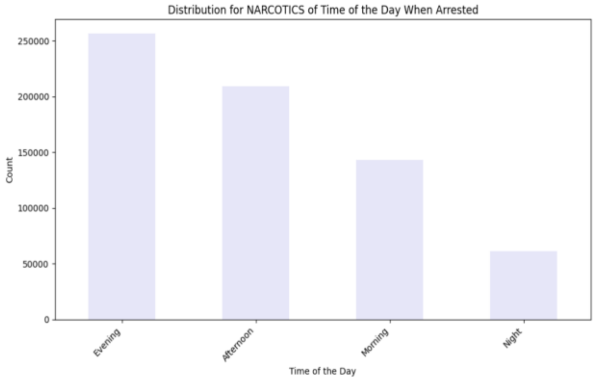

Figure 8: Distribution for narcotics of time of the day when arrested

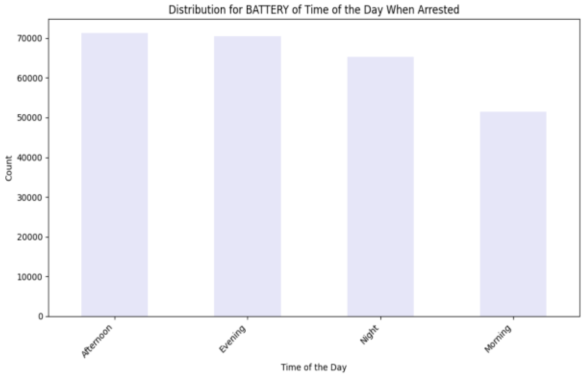

Figure 9: Distribution for battery of time of the day when arrested

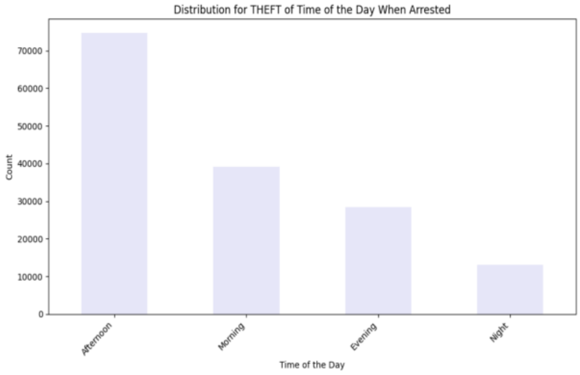

Figure 10: Distribution for theft of time of the day when arrested

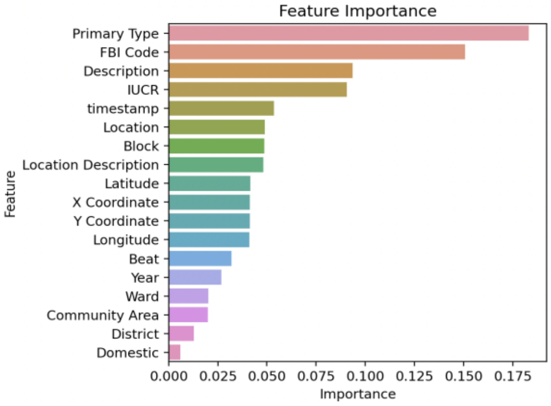

Figure 11: Feature importance to select the most important features

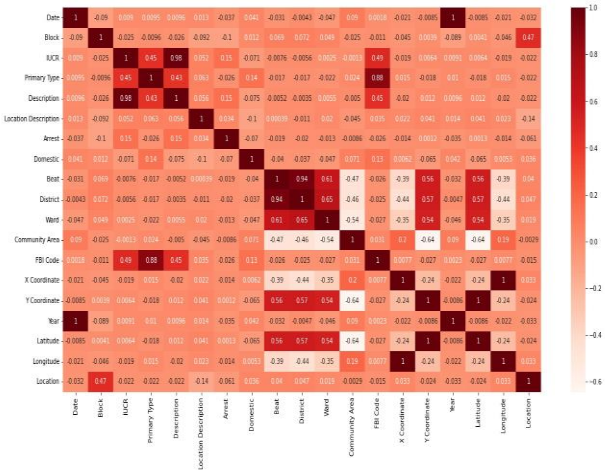

Figure 12: Correlation matrix

Table 1: Feature importance to select the most important features

	Feature	Importance
2	Primary Type	0.183
10	FBI Code	0.151
3	Description	0.094
1	IUCR	0.091
17	timestamp	0.054
16	Location	0.049
0	Block	0.049
4	Location Description	0.048
14	Latitude	0.041
11	X Coordinate	0.041

12	Y Coordinate	0.041
15	Longitude	0.041
6	Beat	0.032
13	Year	0.027
8	Ward	0.020
9	Community Area	0.020
7	District	0.013
5	Domestic	0.006

Table 2: Accuracy and AUC scores for test and validation set for every model

	KNN	Decision Tree	Random Forest	Logistic Regression	AdaBoost	CatBoost	XGBoost	Neural Network
AUC test set	0.8994	0.8996	0.8576	0.8719	0.8974	0.9034	0.9025	0.9029
AUC validation set	0.9039	0.9017	0.8594	0.8786	0.9024	0.9072	0.9055	0.9066
Accuracy test set	0.8804	0.8775	0.8255	0.8620	0.8794	0.8812	0.8802	0.8809
Accuracy validation set	0.8762	0.8749	0.8196	0.8580	0.8767	0.8780	0.8767	0.8772

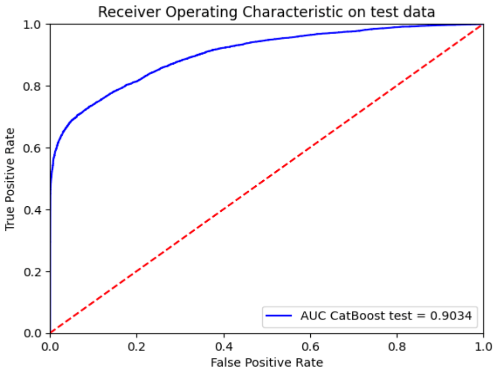

Figure 13: ROC curve CatBoost model

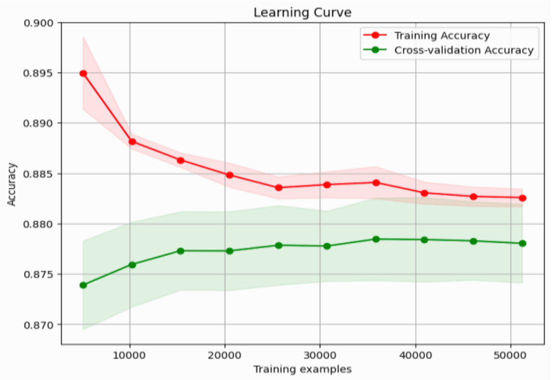

Figure 14: CatBoost Learning Curve with selected features

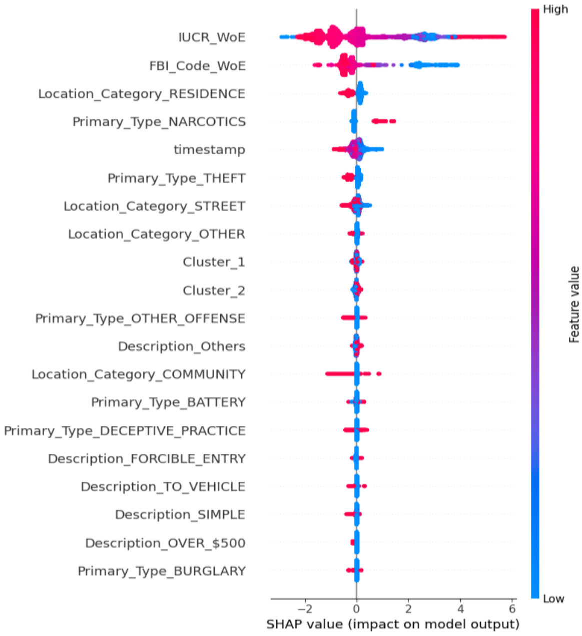

Figure 15: CatBoost Feature importance, generated using SHAP-values

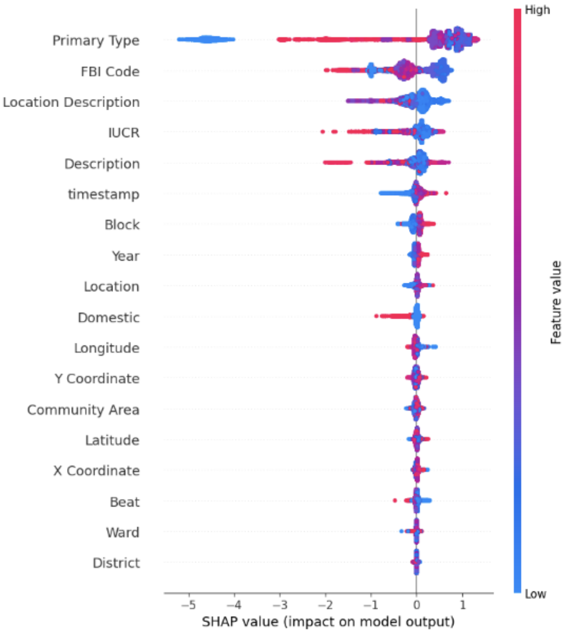

Figure 16: CatBoost Feature importance, generated using SHAP-values (with all datasets, factorized)