DSC 478 Final Project Report

Vatsal Parikh (2058530)

Pooja Mehta (2022333)

Ritu Patel (1938378)

Chicago Crime Analysis

Executive Summary: (Goal, method, conclusion)

This Project Summary presents the goal of Crime prediction developed as a part of DSC-478 final project.

https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2 dataset is utilized to foresee whether reported incidents of crime (with the exception of murders where data exists for each victim) that occurred in the City of Chicago from 2001 to present, minus the most recent seven days. Data is extracted from the Chicago Police Department's CLEAR (Citizen Law Enforcement Analysis and Reporting) system. Each row in the information gives the relevant information about the Crime.

This dataset contains 7.65M observations and 22 attributes.

Before Building our project model we explore and investigate the dataset for a deeper understanding. Firstly, we cleaned the data and found out the total missing values of each column. The Pie chart distribution shows 10 types of different Crime type with variable primary type which has happening recently and with different colors to identify with their percentage ratio and this will give us better idea to proceed with different variables.

The categorical variable COMMUNITY will show the crimes Chicago upon the nearby community. It will show the highest people of the area facing crime. Histogram distribution with Crime rate year with year shows count of crime by its year. Further investigating upon crime rate, we checked the arrested and Domestic Arrested using sub plots via percentage.

Data Preprocessing:

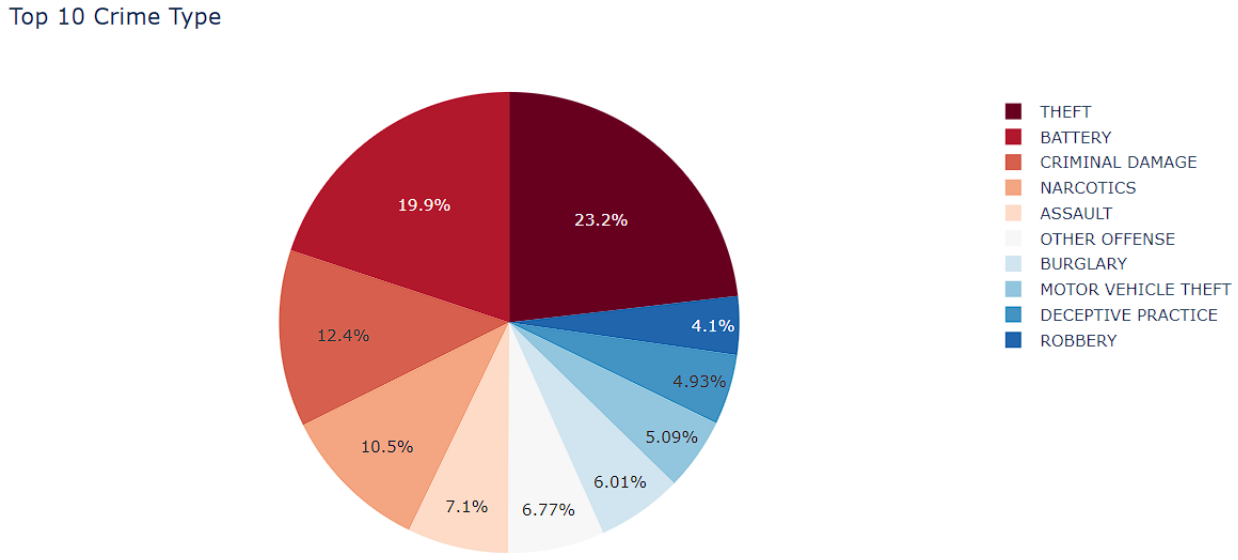
In our dataset "Crimes_-_2001_to_Present.csv", We distinguish between categorical and numerical variables. Along with this, we discovered that there is more column serves no purpose, so we removed it. This cleaned file was used to visualize the data and run machine learning models. We used only 8 columns which are as follows 'Date', 'Primary Type', 'Arrest', 'Domestic', 'District', 'Community Area' and 'Year'.

Community names was extracted from CommAreas.csv for better visualization of data. Then, we transformed Date column in DateTime, so could extract day of the week and hour.

We also had some missing values, but as the attribute is categorical and our Data is huge, we dropped those Null values.

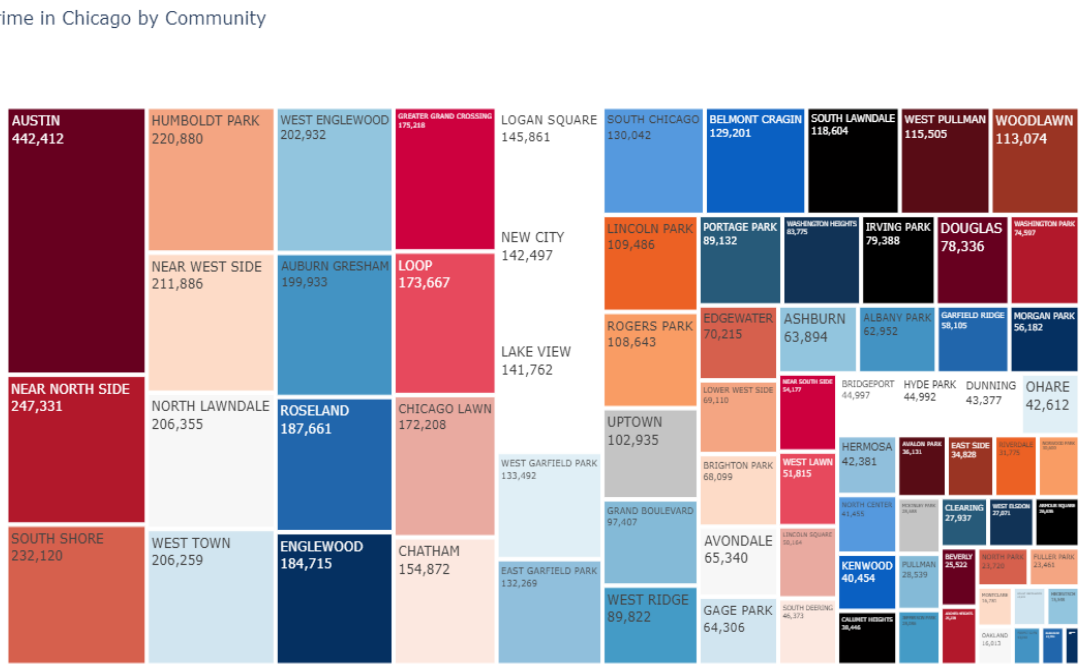
Data Visualization:

First of all, we have to find out that total how many percentages of crime and which type of crime happens in Chicago, for that we use "Primary Type" column, and we generate top 10 crime type of pie chart so, we can easily visualize that which crime has a more percentage in Chicago.



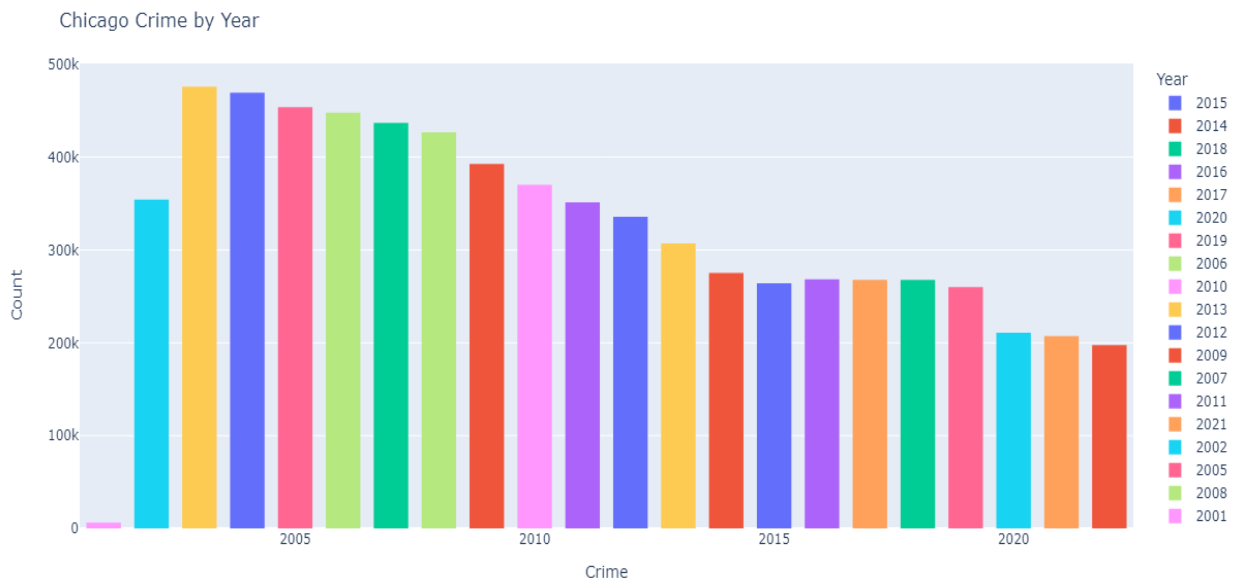
From above pie chart, we can clearly say that this represents Top 10 crime type and “Theft” crime has more 23.2% compared to other crimes. Also, “Battery” type has a 20% and “Narcotics” has 10.5%. this pie chart helps for better visualization of crime type.

In addition, we want to find out that which community has more crime happen in our dataset.. For visualization, we used Tree map plot. Because Tree map shows all values and labels in different box with different colors, which helps to easily visualize.



From above Tree map, we can see that it shows community name with their value numbers which shows crime happens that area. For example, Austin community has 442,412 value crime happens, which is the highest number of crimes happen in Austin community compared to other community.

Moreover, to find out that in which year has more crime happen in Chicago city. For that we used Histogram plot for better visualization. This plot shows from year “2001” to “2022” crime reports by their counts. We used histogram because it shows in different colors for each year so we can identify that which year has more crime rate than other years.



From this plot, we can easily visualize that we took “Crime” for x-axis and “Count of crime” in Y-axis. Also “Year” shows in different colors. In the Year, 2001 has very low count of crime happen which is only 6246 and it shows in light pink color. On other side, year 2003 has more count of crime happen which is like 475.917k, and it shows in yellow color. As result, we can say that Year between 2001 to 2010 has more crime happen in Chicago. By comparing other years, between 2011 to 2022 has less crime happen in Chicago.

Data Modeling:

From, our dataset and as from above pie chart of crime type, we have top 10 crime primary types. So, we decided to do classification on major crimes and minor crimes. But due to high percentage of data falling into major crimes, we got high accuracy from the very start. Hence, we dropped that planed and we decided to move in specific crime type for specialized assistance which can provided.

The idea is, if we could classify the crime based on day of the week, hour of the day and place of incident, we can provide real-time assistance to the impacted victim without wasting time.

Predicting if the crime requires medical assistance:

First application of it would be providing medical expertise. After reaching the crime scene, if we discover that victim requires medical expertise, then calling for assistance could waste a lot of time and could endanger the victim.

So, we started with two crimes, Assault and Battery, which could require medical assistance for the victim. We are going to classify the crime in real time and thus predict if medical assistance is needed.

We are going to use District, Year, Hour, Community and Day for model creation. As we are predicting in real world data, Arrest attribute does not apply here. We can classify if the crime is Domestic or not in our model, but we can see from pie charts of the given data, Domestic category makes up for very small percentage of data and thus it would be really difficult to classify.

After creating dummy variables, Primary_med is set as target variable, and train/test split is done on the data. 10% of data can be set as test set here, due to size and variability in the data. But to not overfit the data, we will do our analysis on 20% test set.

Decision Tree:

Decision Trees (DTs) are a type of non-parametric supervised learning method that can be used for classification and regression. The model we will be using for this classification is Decision Tree. We get accuracy of 0.742 on default parameters. We also tried on gini index but received same results.

We also used one ensemble method **Random Forest Classifier**, so we can compare performance of standard decision tree against some **ensemble methods**. We received the accuracy of 0.737, which is not an improvement but maybe after fine tuning the parameters we can achieve even higher accuracy. We selected best model parameters after **exploring different parameters** like min_samples_leaf, m_depth and max_depth. After fine tuning the parameters using **cross validation**, running our final model with these parameters. We got an accuracy of 0.752. Which is not a huge jump again, but based on the attributes we selected for model creation this is best accuracy we could get.

We can't say this can be used in real world just yet. More optimization and tuning is required, also using different models can also improve accuracy drastically.

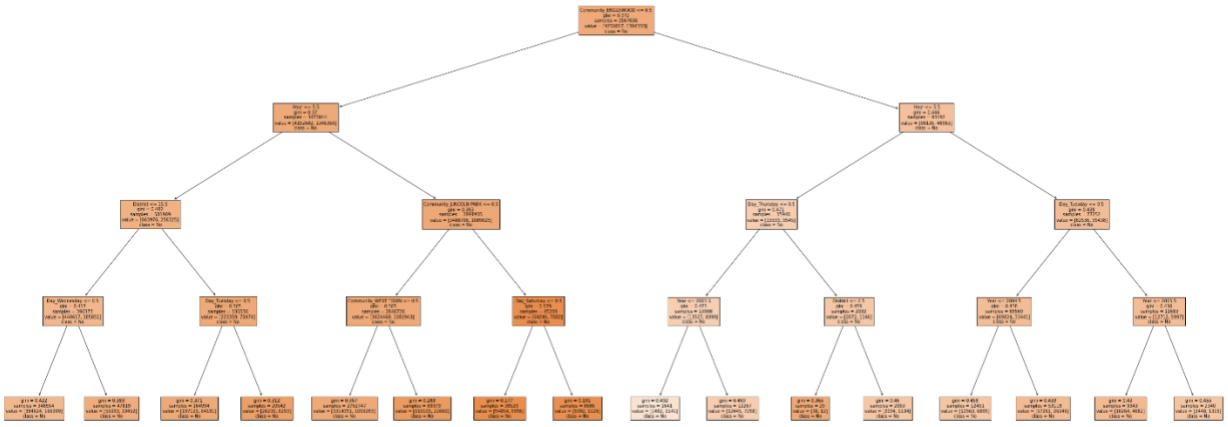
For some values of test data, out model is not able to predict the class, therefore while creating classification report, some values give warning regarding divide by zero. Therefore, we left the classification report and just focused on accuracy.

```
print("Accuracy on Training: ",  rf.score(crime_train, target_train))
print("Accuracy on Test: ",  rf.score(crime_test, target_test))
```

```
Accuracy on Training:  0.7529268487108403
Accuracy on Test:  0.7519173974412027
```

We can see here that both train and test accuracy are very close and thus there doesn't seem to any overfitting by the model on data.

Plotting Final Decision Tree:



Predicting if the crime requires DEA support:

Second application of it would be identifying if the crime is related to Narcotics. DEA specializes in crime related to narcotics, and sometimes not handling the crime scene properly could erase some of the evidence. So, to include specialized task force of the field makes complete sense. We started Narcotics in Primary Type attribute and will be using the same District, Year, Hour, Community and Day for model creation. Also, the same logic applies here for attribute selection here as above model. After creating dummy variables, Primary_narco is set as target variable, and train/test split is done on the data.

Naive Bayes:

The Nave Bayes Classifier is a simple and effective Classification algorithm that enables in the development of fast machine learning models capable of making quick predictions. Using this method, we got the score of training and test set of "NARCO" primary type of crime. Below, image we can see that the score of training set is **"0.8986",** and score of test set is **"0.8989".**

```
nbclf = naive_bayes.MultinomialNB()
nbclf = nbclf.fit(crime_train, target_train)
print("Score on Training: ", nbclf.score(crime_train, target_train))
print("Score on Test: ", nbclf.score(crime_test, target_test))
```

```
Score on Training:  0.8986845345119262
Score on Test:  0.8989766794587639
```

We are getting very good accuracy here, but we can try different models and see which one fits best.

LDA [LINEAR-DISCRIMINANT-ANALYSIS]:

Linear discriminant analysis (LDA) is a type of linear combination, which is a numerical process that uses multiple data items and applies functions to that set to analyze multiple classes of objects or items separately. we can see that the score of training set is **"0.9026",** and score of test set is **"0.9027".** which gives best accuracy value compared to naïve bayes.

```
ldclf = LinearDiscriminantAnalysis()
ldclf = ldclf.fit(crime_train, target_train)
print("Score on Training: ", ldclf.score(crime_train, target_train))
print("Score on Test: ", ldclf.score(crime_test, target_test))
```

```
Score on Training:  0.9026410281391991
Score on Test:  0.9027647085499203
```

DECISION TREE:

The goal is to build a model that predicts the value of a target variable using simple decision rules derived from data features. we can see that the score of the training set is **"0.9058",** and the score of the test set is **"0.9037".** which gives better and more accurate accuracy values compared to naïve Bayes and LDA.

```
treeclf = tree.DecisionTreeClassifier(criterion='entropy', min_samples_split=3)
treeclf = treeclf.fit(crime_train, target_train)
print("Score on Training: ", treeclf.score(crime_train, target_train))
print("Score on Test: ", treeclf.score(crime_test, target_test))
```

```
Score on Training:  0.9058246459350061
Score on Test:  0.9037545844646876
```

Conclusion:

Our model gives very high accuracy using default parameters in different models. Further accuracy could be achieved here by fine-tuning the parameters using cross-validation. We can also see that Train and Test accuracy both are really close thus there doesn't seem to be any overfitting either.

This model can be used in real life to assist the police and reduce the crime rate related to Narcotics and DEA assistance can also help retain evidence at the crime scene.