Crimes in Chicago Report

Purdue University

CNIT 559

Spring 2018

Benya Chongolnee
Scott Jean
Nidhi Sakhala

Introduction:

Crimes happen everywhere no matter how large or educated the population. What makes crimes different is the amount of crimes that do happen and where. Chicago is relatively close to Purdue University and that is the location where some of the students may relocate to after graduation. It is very important to ensure that where the students relocate to is in a safe and secure neighborhood. With that being said, we found a dataset from Crimes - 2001 to present | City of Chicago | Data Portal. (n.d.). This dataset is an extensive dataset of crimes in Chicago from 2001 to 2018 provided by the City of Chicago. This dataset has 1,048,575 instances with 22 attributes. The attributes includes ID, Case Number, Date, Block, IUCR (Illinois Uniform Crime Reporting code), Primary Type, Description, Location Description, Arrest, Domestic, Beat, District, Ward, Community Area, FBI Code, X Coordinate, Y Coordinate, Year, Updated On, Latitude, Longitude, and Location. This report will use decision tree classification, which is a data mining method to help students see which type of crimes (primary type) will most likely occur in their desired neighborhood.

Research Process:

For this research, we will only be using Block, Year, Description, Location Description, District, Ward, Community Area, Latitude, Longitude, and Primary Type attributes.

Attribute Definitions: (Crimes - 2001 to present | City of Chicago | Data Portal., n.d.)
Block - The partially redacted address where the incident occurred, placing it on the same block as the actual address.
Year - Year the incident occured.
Description - The secondary description of the IUCR (Illinois Uniform Crime Reporting) code, a subcategory of the primary description.
Location Description - Description of the location where the incident occurred.
District - Indicates the police district where the incident occurred.
Ward - The ward (City Council district) where the incident occurred.
Community Area - Indicates the community area where the incident occurred. Chicago has 77 community areas.
Latitude - The latitude of the location where the incident occurred. This location is shifted from the actual location for partial redaction but falls on the same block.
Longitude - The longitude of the location where the incident occurred. This location is shifted from the actual location for partial redaction but falls on the same block.
Primary Type - The primary description of the IUCR (Illinois Uniform Crime Reporting) code.

We use the first nine attributes (Block, Year, Description, Location Description, District, Ward, Community Area, Latitude, Longitude) to predict which type of crimes (Primary Type) would likely happen. By accurately predict the crime type, it will be beneficial for students to figure out which part of Chicago they should relocate to. Some attributes that is provided in the dataset such as ID and case number are not necessary when deciding what parts of Chicago is safe. Having these unnecessary attributes will results in the data being less accurate.

The dataset includes the following types of attacks (Primary Type):
'ARSON', 'ASSAULT', 'BATTERY', 'BURGLARY', 'CONCEALED CARRY LICENSE VIOLATION', 'CRIME SEXUAL ASSAULT', 'CRIMINAL DAMAGE', 'CRIMINAL TRESPASS', 'DECEPTIVE PRACTICE', 'GAMBLING', 'HOMICIDE', 'HUMAN TRAFFICKING', 'INTERFERENCE WITH PUBLIC OFFICER', 'INTIMIDATION' 'KIDNAPPING', 'LIQUOR LAW VIOLATION', 'MOTOR VEHICLE THEFT', 'NARCOTICS', 'NON - CRIMINAL', 'NON-CRIMINAL', 'NON-CRIMINAL,' 'OBSCENITY', 'OFFENSE INVOLVING CHILDREN', 'OTHER NARCOTIC VIOLATION', 'OTHER OFFENSE', 'PROSTITUTION', 'PUBLIC INDECENCY', 'PUBLIC PEACE VIOLATION', 'RITUALISM', 'ROBBERY', 'SEX OFFENSE', 'STALKING', 'THEFT', 'WEAPONS VIOLATION'

Phase 1: We decided to use the decision tree method to predict the types of crimes in the future. We do this by using 70% of our data set to train the model for our problem. After getting this model, we then use the model to predict the other 30% of our data set to see how accurate the model is. We believe that decision tree is the best method when using categorical variables and when predicting new values. We decided to use this data mining approach after figuring out three advantages and disadvantages of the different data mining approaches described in Figure 1. Decision tree is the most appropriate method for our dataset because it can handle categorical data (primary types, description, etc). However, decision tree does not easily handle non-numeric data, thus we need to convert categorical data into numeric data. Overall, decision tree classification method helps create decision rules that easily predicts crimes in Chicago.

Data mining Approaches	**Description**	**Advantages**	**Disadvantages**
Decision Tree- Classification Learning (Fakhari, 2013)	Method used to create a model that predicts the value of a variable by using decisions rules (if-then statements).	-Handles continuous and discrete data - Deals with noisy and incomplete data - Is good for categorical data	-High classification error rate when training set is small. - Calculations grow while problem grows - Need to discrete data for some datasets - Does not easily handle non-numeric data
Linear Regression (Halthor, 2017)	It creates a model of the relationship between a dependent variable and one or more explanatory variables.	- It is a very simple method - Helps see any outliers - Works with most cases	- Assumes linearity in the dataset - It assumes there is a straight-line relationship between dependent and independent variables - If a number goes out of range the data becomes distorted
K-Means- Clustering (Gulyani, 2013)	K-means is a partitioning clustering method where the center of the cluster is represented by the mean of the objects in the cluster.	- When *k* is small it can be computed faster than hierarchical methods. - produces tighter clusters	- Need more time for some tasks - Does not work well with the global clusters - Difficult to find the actual value of *k*.

Figure 1: Data mining approaches

Phase 2: Decision tree is a model where it breaks down the dataset into small subsets or category that are common with each other. The decision tree keeps breaking down the data until it is unable to find a better fit. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches where it could lead to a different decision node or a leaf node. Leaf node represents a classification or decision that cannot be broken down into another decision node (Decision Tree - Classification, n.d.). Decision trees can handle both categorical and numerical data which is very important in our dataset. After deciding that decision tree algorithm is the best algorithm to use for our problem, we started looking at ways to integrate the algorithm to our dataset. We started by coding the algorithm using Python since it was the easiest language to perform our test. The code is in Appendix A. The code includes confusion matrix, accuracy scores, and the decision tree itself. This decision tree is attached to this document as a pdf file. We used the scikit-learn library of Python which provides the DecisionTreeClassifier function. We use both the gini index and the entropy metrics to build the tree to see how it affects our accuracy. We also vary the depth of the tree and the minimum number of samples for each leaf node and map the difference in results. The important parameters of the decision tree classifier are-

Criteria: The function to measure the quality of a split.

There are 2 types of criterias generally used - gini index and entropy.

1. $Gini : Gini(E) = 1 - \sum_{j=1}^{c} p_j^2$
2. $Entropy : H(E) = - \sum_{j=1}^{c} p_j \log p_j$

Max_depth: The maximum depth of the tree.

Min_Samples_leaf: The minimum number of samples required to be at a leaf node.

Results:

Running the python code gives the results that can be seen in Appendix B. Running the code gives the confusion matrix and the accuracy of the model which is used to predict the type of crime based on the other location factors. First, the accuracy of this test is 90.525544107. This means that our model was around 90% accurate when predicting the type of crime for the test data. This is incredibly good for someone who may be thinking about moving into Chicago. They are can figure out if their desired neighborhood may have a chance of a particular type of attack or if it is a safe location to move into. It can also help them be better prepared for their safety.

The next part of the result is the confusion matrix. The confusion matrix shows which variables were predicted right or wrong and how many. The multi-class confusion matrix gives a numerical summary of the number of data instances for each type of attack and what the instance has been predicted as. It is visually represented as a table.

		Prediction				
		Class 1	Class 2	Class 3	...	Class n
Actual	Class 1	**Accurate**				
	Class 2		**Accurate**			
	Class 3			**Accurate**		
	...				**Accurate**	
	Class n					**Accurate**

(WSO2 Documentation, n.d.)

As seen in the figure, the diagonal elements show the attacks that have been predicted correctly by the model. The other cells display the number of attacks which are actually an attack of the 'actual' class row but have been predicted wrongly as the class under prediction column. To explain these results in the context of our dataset, it will be difficult to read the whole 33 by 33 matrix in a single visualization, so we will explain it here by showing just one row of the confusion matrix. This row shows us the distribution of crime type-theft. The numbers represent that out of all the thefts in our testing set, how many were predicted as each of the crime type. So we can see that 66017 thefts were correctly identified but 45 were predicted as battery, 30 as burglary, 476 as 'interference with public officer' and 47 as sex offense. These misclassified instances reduce the accuracy of our model.

Predicted Class	Number of Instances	Predicted Class	Number of Instances
ARSON	0	'NARCOTICS'	0
ASSAULT	0	'NON - CRIMINAL'	0
'BATTERY'	45	'NON-CRIMINAL'	0
'BURGLARY'	30	'NON-CRIMINAL'	0
'CONCEALED CARRY LICENSE VIOLATION'	0	'OFFENSE INVOLVING CHILDREN'	0
'CRIME SEXUAL ASSAULT'	0	'OBSCENITY'	0
'CRIMINAL DAMAGE'	0	'ROBBERY'	0
'CRIMINAL TRESPASS'	0	'OTHER OFFENSE'	0
'GAMBLING'	0	'PROSTITUTION'	0
'DECEPTIVE PRACTICE'	0	'PUBLIC INDECENCY'	0
'HOMICIDE'	0	'RITUALISM'	0
'HUMAN TRAFFICKING'	0	'PUBLIC PEACE VIOLATION'	0
'INTERFERENCE WITH PUBLIC OFFICER'	476	'OTHER NARCOTIC VIOLATION'	0
'INTIMIDATION'	0	'SEX OFFENSE'	47
'KIDNAPPING'	0	'STALKING'	0
'LIQUOR LAW VIOLATION'	0	'THEFT'	66017
'MOTOR VEHICLE THEFT'	0	'WEAPONS VIOLATION'	0

Problems:

We ran into a few problems while analyzing the data. The first problem we ran into was that running the code took longer than expected. We realized very early on that it was because of the null values that was in the dataset. We debated about using the mean or the median to replace the null values. However, it does not make sense to do this for Community area or latitude/longitude because it means that we are placing non-existing crimes in the dataset, that is not accurate. Instead, we decided to completely remove all of the data that has null values. Because of this, we went from having 1,048,575 instances to 991,225.

Another problem that we ran into was the need to convert all of the categorical variables into numerical variables. Because decision trees do not handle non-numerical data very well, we decided to change our categorical variables into numbers. For example, in location description, street is changed to ‘1’, apartment is ‘2’, and so on. Doing this will help running the decision tree algorithm easier even though the numerical values is not comparable, it helps categorize each categorical variables into numbers.

Another problem we ran into was figuring out which attributes was necessary in predicting the crimes type. At first, we used some attributes in our code to predict the primary types. We noticed that doing this gave us a very bad accuracy score. With this realization, we wants to only use attributes that is *not* unique and has a relationship with the crimes. Having a unique attributes such as ID will not do anything in predicting new data set, mainly because each ID is unique to a crime. There could not be crimes that have the same ID. From this, we narrowed our attributes to the nine attributes that is beneficial in predicting the crime types.

Discussion:

We found that some attributes either do nothing in accurately predict the data or reduce the accuracy. Initially, we thought that the more attributes we have in the test, the better the accuracy will be. This is found to be the opposite. The attributes we used were Description, Block, Year, Location Description, District, Ward, Community Area, Latitude, and Longitude attributes. If we add other attributes such as community area or FBI code, the accuracy plummeted; therefore, the attributes we used were the best in accurately predicting the rest of the dataset. This can be caused from many different issues. For example, the attributes may be unique to the specific crime. By having that in the model, future crimes will be harder to predict since the computer will not know what to do with it.

The results we found for this report can benefit more people than just students. This test can help the City of Chicago figure out which area has a high crime rate of what type. Doing so can tell the City of Chicago which crime type is a problem so that they can work their way into reducing this alarming rate. The results we found will also be useful if it was displayed visually. By having a visualization about crimes in the City of Chicago, audience everywhere will be able to better understand the dataset given. It can help identify patterns or trends that may be associated with the crimes in Chicago.

Another way that this dataset can be useful is to use it to predict the block (apartment, street, sidewalk, etc). By knowing the block of a crime, it can help people to know which block to avoid or to be extra precautious. This can even help the City of Chicago to ensure that there will be more police in that area in the case that a crime were to occur. There are many analysis that could be done in the dataset that we found; it is just a matter of what problems the City of Chicago find the most important.

One interesting analysis that we did was to vary the training metrics of the decision tree and record the variation in accuracy. These were the variations that we saw. We concluded that the accuracy improved with more depth of the tree. Varying the decision parameter from entropy to gini did not matter much, nor did changing the minimum values needed at leaf node.

Criteria	**Max_depth**	**Min_leaf**	**Accuracy**
entropy	3	5	49.38
entropy	3	100	49.382
entropy	3	50	49.381
entropy	10	100	91.7
entropy	10	50	91.728
gini	3	50	49.651

gini	3	100	49.651
gini	10	50	90.746
gini	10	100	90.716

Conclusion:

We chose a dataset of over a million instances of crime from the City of Chicago. Our goal is to provide a way for a student to predict where they may want to live and avoid areas of crime. We discussed the advantages and disadvantages of different data mining approaches such as decision tree, linear regression, and partitioning method clustering/K-means. With that being said, we chose to use the decision tree algorithm, which uses decision rules called if-then statements to predict the values of the variables in our dataset. In order to run the decision tree algorithm, we used Python's scikit learn library to code our algorithms. We varied the model architecture and recorded the variation in the accuracy. This Python code includes the final model with its confusion matrix, accuracy scores, and the decision tree itself. Our model was around 90% accurate when predicting the type of crime for the test data. This is a very good prediction when we are trying to better understand the Chicago crime rates and its associated location.

References:

Crimes - 2001 to present | City of Chicago | Data Portal. (n.d.). Retrieved April 11, 2018, from https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2

Data Mining Classification & Prediction. (2018, January 08). Retrieved April 11, 2018, from https://www.tutorialspoint.com/data_mining/dm_classification_prediction.htm

Decision Tree - Classification. (n.d.). Retrieved April 17, 2018, from http://www.saedsayad.com/decision_tree.htm

Fakhari, A. (2013, April 15). What are pros and cons of decision tree versus other classifier as KNN,SVM,NN? Retrieved April 16, 2018, from https://www.researchgate.net/post/What_are_pros_and_cons_of_decision_tree_versus_other_classifier_as_KNN_SVM_NN

Gulyani, T. (2013, February 14). Technology For You. Retrieved April 17, 2018, from http://playwidtech.blogspot.com/2013/02/k-means-clustering-advantages-and.html

Halthor, A. (2017, November 25). What are the advantages and disadvantages of linear regression? Retrieved April 16, 2018, from https://www.quora.com/What-are-the-advantages-and-disadvantages-of-linear-regression

MP, A. (2015). Tools Pros and Cons of Clustering Algorithms using Weka Tools. *International Journal of Computer Applications*. Retrieved April 11, 2018.

Trevino, A. (n.d.). Introduction to K-means Clustering. Retrieved April 11, 2018, from https://www.datascience.com/blog/k-means-clustering

WSO2 Documentation. (n.d.). Retrieved April 17, 2018, from https://docs.wso2.com/display/ML110/Model Evaluation Measures

Yang, H. (1970, January 01). Heron's Note. Retrieved April 11, 2018, from http://note.heron.me/2013/04/applying-k-mean-on-csv-files-using.html

Appendix A:
Code in Python

```
# coding: utf-8

      # In[3]:


      import numpy as np
      import pandas as pd
      from sklearn.metrics import confusion_matrix
      from sklearn.cross_validation import train_test_split
      from sklearn.tree import DecisionTreeClassifier
      from sklearn.metrics import accuracy_score
      from sklearn.metrics import classification_report
      from sklearn import tree

      # In[4]:


      def importdata():
          #print ("in import function")
          crime_data=pd.read_csv('dwh.csv',sep=',',header=0)
          #print ("database length: ", len(crime_data))
          print ("database shape: ", crime_data.shape)
          #crime_data=crime_data[(crime_data.Type == "BURGLARY") |
      (crime_data.Type == "ASSAULT") | (crime_data.Type == "MOTOR VEHICLE
      THEFT")]
          crime_data=crime_data.dropna()
          #print ("Crime types chosen - Burglary, Assault, Motor vehicle
      theft")
          print ("New database length: ", len(crime_data))
          index=0
          for val in crime_data['Block']:
              crime_data['Block'][index] = val[6:]
              index=index+1
              #print (index)
          return crime_data


      # In[11]:


      def convertdata(crime_data):
          for feature in crime_data.columns:
              if (crime_data[feature].dtype == 'object'):
```

```
            crime_data[feature] =
pd.Categorical(crime_data[feature]).codes
    print ("data converted to numeric values")
    return crime_data

def splitdata(crime_data):
    X=crime_data[['Block','Description','Location
Description','District','Ward','Community
Area','Latitude','Longitude','Year']].copy()
    X=convertdata(X)
    X=X.as_matrix()
    Y=crime_data.values[:,5]


    X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size = 0.3, random_state = 100)
    print ("Features used - Block, Description, Location
Description, District, Ward, Community Area, Latitude, Longitude")
    print ("data split into training and testing sets")
    return X,Y,X_train,X_test,Y_train,Y_test


# In[23]:


def trainentropy(X_train, Y_train):
    clf_entropy = DecisionTreeClassifier(criterion = "entropy",
random_state = 100,max_depth = 10, min_samples_leaf = 1000)
    clf = clf_entropy.fit(X_train, Y_train)
    tree.export_graphviz(clf, out_file='tree5.dot')
    print ("classifier trained")
    return clf


# In[24]:


def prediction(X_test,clf_object):
    y_pred = clf_object.predict(X_test)
    #print ("prediction values: ",y_pred)
    return y_pred


# In[25]:
```

```
def calaccuracy(y_test,y_pred):
    print ("Confusion Matrix: ")
    conf = confusion_matrix(y_test, y_pred)
    for k in conf:
        print (k)
    print ("Accuracy : ",accuracy_score(y_test,y_pred)*100)
    #print ("Report : ")
    #print (classification_report(y_test, y_pred))


# In[27]:


def main():
    data=importdata()
    x,y,xtrain,xtest,ytrain,ytest=splitdata(data)
    clf=trainentropy(xtrain,ytrain)
    ypred=prediction(xtest,clf)
    calaccuracy(ytest,ypred)


# In[28]:


if __name__=="__main__":
    main()
```

Appendix B:
Results

```
database shape:  (1048575, 22)
New database length:  991225
data converted to numeric values
Features used - Block, Description, Location Description, District, Ward,
Community Area, Latitude, Longitude, Year
data split into training and testing sets
classifier trained
Confusion Matrix:
[   0    0   93   80    0    0    0    0  354    0    0    0    0    0    0    0    0    5
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0]
[     0   7794  12751      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0]
[     0   5025  51711      0      0      0      0      0      0      0      0      0
      7      0      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0]
[     0      0      0  15238      0      0      0      0    260      0      0      0
      0      0      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0]
[  0   0  16   0   0   0   0   0   0   0   0   0   3   0   0   0   0   0   0   0   0   0   0   0
0
   0   0   0   8   0   0   0   0]
[   0  265  179   83    0  802    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0]
[     0      0      0      0      0      0  34403      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0      0      0
     32      0      0      0      0      0      0      0      0]
[    0     0     0     0     0     0   821  7207     0     0     0     0     0     0
0
     0     0     0     0     0     0     0     0     0     0     0     0     0     0
0
     0     0     0]
[     0      0    113      0      0      0      0      0  13097      0      0      0
      0      0      0      0    433      1      0      0      0      0      0      0
     18     94      0      0      0     51      0     49     38]
[   0    0    0    0    0    0    0    0   13    0    0    0   18    0    0    0    0    0
    0    0    0    0    0    0    3    0    0    0    0    0    0  385    0]
[   0    0    0    0    0    0    0    0  204    0    0    0    1    0    0    0    0    0
    0    0    0    0    0    0    1    0    0    0    0    0    0    0    0]
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 3 0 0 0 0 0 0 0 0]
[   0    1    0    0    0    0    0    0    4    0    0    0  894    0    0    0    0    0
    0    0    0    0    0    0   49    0    0    0    0    0    0    0    0]
[   0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
   0   0   0   0   3   0 170   0   0   0   0   0   0   0   0]
 [  0   0  12   0   0   0   0   0   4   0   0   0   0   0   0   0   0   0
   0   0   0   0  84   0  37   0   0   0   0   0   0   0 145]
 [  0   0   0   0   0   2   0   0 222   0   0   0   0   0   0   0   0   0
   0   0   0   0   0   0 117   0   0   0   0   0   0  70   0]
 [    0     0     0     0     0     0     0     0     0     0     0     0
      0     0     0     0 12572     0     0     0     0     0     0     0
      0     0     0     0     7   226     0     0     0]
 [    0     0     0     0     0     0     0     0   114     0     0     0
      0     0     0     0     0 21384     0     0     0     0     3     0
      0   288     0     0    75    15     0    22     0]
 [0 0 0 0 0 0 0 0 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [ 0  0  0  0  0  1  0  0  2  0  0  0  0  0  0  0  0 22  0  0  0  0  0  0
 0
   0  0  0  0  0  0  0  0]
 [0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]
 [ 0  0  0  0  0  0  0  0  0  0  0  0 36  0  0  0  0  0  0  0  0  0  0  0
 0
   0  0  0  0  0  0 12  0]
 [  0   0 500   0   0   0   0   0 118   0   0   0   0   0   0   0   0   5
   0   0   0   0 960   0 490   0   0   0   0   0   0  43   0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0]
 [    0     7    47     0     0     0     0     0    14     0     0     0
    109     0     0     0     5    18     0     0     0     0     2     0
  18755     0     0     0   179     0     0     0     0]
 [   0    0    0    0    0    0    0    0    0    0    0    0    0    0
 0
     0    0   48    0    0    0    0   62    0   31 1791    0    0    0
 0
     0    0    0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 9 0 0 0 0 0 0 0 0]
 [   0    2   11    0    0   38    0    0  233    0    0    0    0    0
 0
     0  155    0    0    0    0    0    0    0  272    0    0 1219    0
 0
     0    3    0]
 [    0     0   645     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0     0     0     0     0     0     0
    273     0     0     0 11349     0     0     0     0]
 [  0  52 435   0   0   0   0   0   0   0   0   0  37   0   0   0  51   0
   0   0   0   0   0   0  71   0   0   0   0 429   0   0   0]
 [ 0 52 92  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 9
   0  0  0  0  9  0  0  0]
 [    0     0    45    30     0     0     0     0     0     0     0     0
```

```
     0     0     0     0     0   476     0     0     0     0     0     0
     0     0     0     0     0    47     0 66017     0]
[    0     0     0     0     0     0     0     0     0     0     0     0   216     0
0
     0     0    68     0     0     0     0     0     0     0     0     0     0     0
1
     0     0 3572]
Accuracy :  90.525544107
```