**1.SYNOPSIS**

Crimes occur everywhere around us, affecting the quality of life and growth of the economy of a society. Although crimes occur everywhere, criminals target familiar areas for them whenever they get the opportunity. Thus, given a data mining approach, it is possible to process data such as the crime type, date, area and so on to find a pattern among these crimes. This could help us predict whether the area is safe or unsafe, for a given time, day and area.

By this, we hope to raise the awareness among people about their surroundings and help them be aware and undertake safety measures at the dangerous time periods. This could also be useful to the police department of the area so they can send suitable police force and maintain tight security in crime prone areas and times.

Our Data Mining assignment aims at performing Tactical Crime analysis on real-world crimes dataset of Los Angeles in California from 2010 to 2015. To extract frequent patterns of crime on the dataset, we used several techniques and algorithms to generate various graphs to help us analyze the data and be able to predict the crime rates in specific locations at a particular time. We have provided statistical analysis of different crime types with their demographic information.

*About the Dataset:*  
Our dataset initially consisted of several attributes such as date reported, date occurred, time occurred, area id, area name, reporting district, crime code, crime code description, victim age, victim sex, victim descent, weapon used code, weapon description, status code, status description, cross street, location.  
But for the purpose of our assignment, we narrowed them down to 12 attributes. A detailed description of each attribute is given below.

**2.METHODOLOGY**

Identifying relationships and patterns between the various attributes involved and the crimes committed could significantly facilitate the estimation of potential crime hotspots.

A representation of the process involved in the prediction process is shown in Figure 2.1.

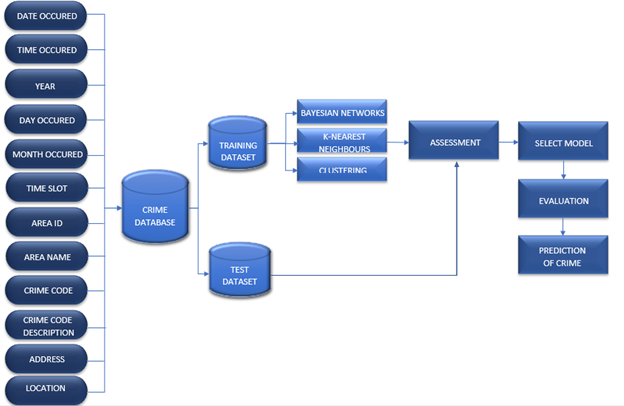


FIGURE 2.1. PROCESS INVOLVED IN PREDICTION PROCESS

**2.1 Data Preprocessing**

Under ‘Assessment’, a few data preprocessing operations mentioned below were performed to simplify the processing of the vast and varied nature of the data set taken into consideration.

* ***Data Reduction***

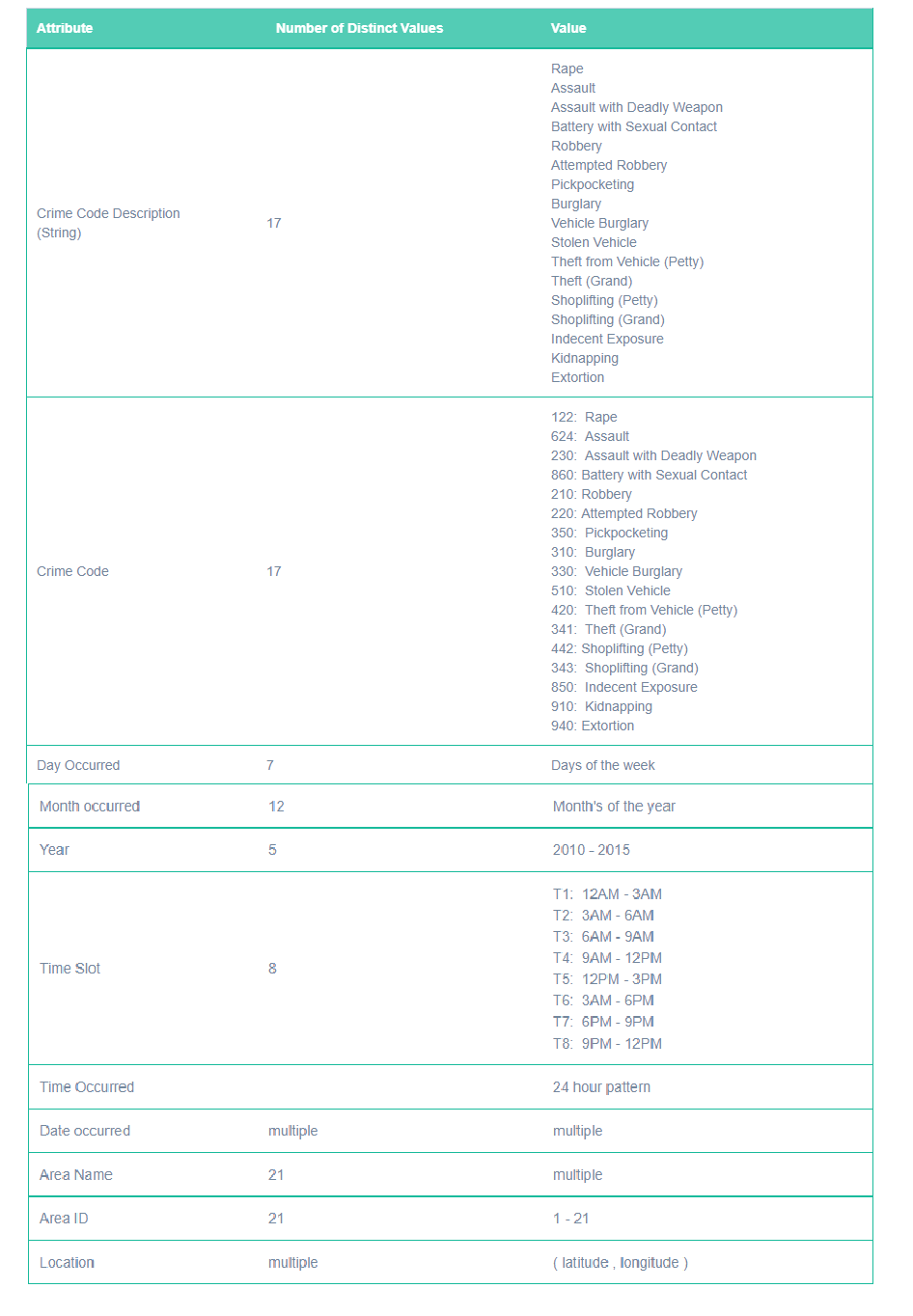
Since the available data set has a plethora of instances, data reduction was performed. Reduction of dimensionality was implemented by taking only about 5 of the total 12 attributes available into consideration. All other redundant and irrelevant attributes were eliminated from the data set.

* ***Data Integration***

Attributes Time Slot and Time Occurred; Area Name and Area ID were unified under attributes Time Slot and Area Code respectively to avoid different attribute naming.

* ***Data Transformation***

In order to identify the patterns more efficiently and for increase the accuracy of the models, 8 time slots and 21 area codes were devised.

FIGURE 2.2. DATA SET ATTRIBUTES TABLE

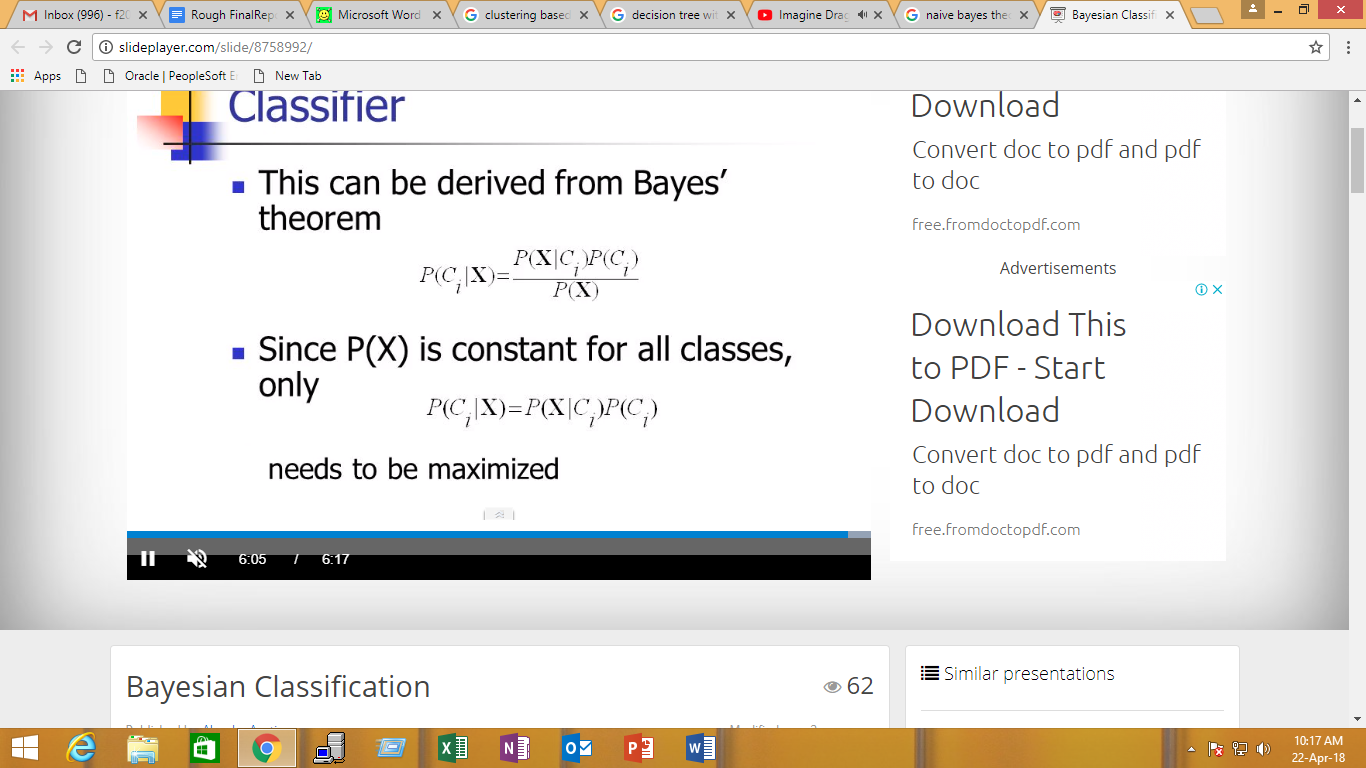
**2.2 Data Analysis**

In order to deduce conclusive patterns from the crimes that were committed in Los Angeles, Naive Bayesian classifier, Decision Tree with KNN filter applied and Density based Clustering algorithms were applied on the available data set.

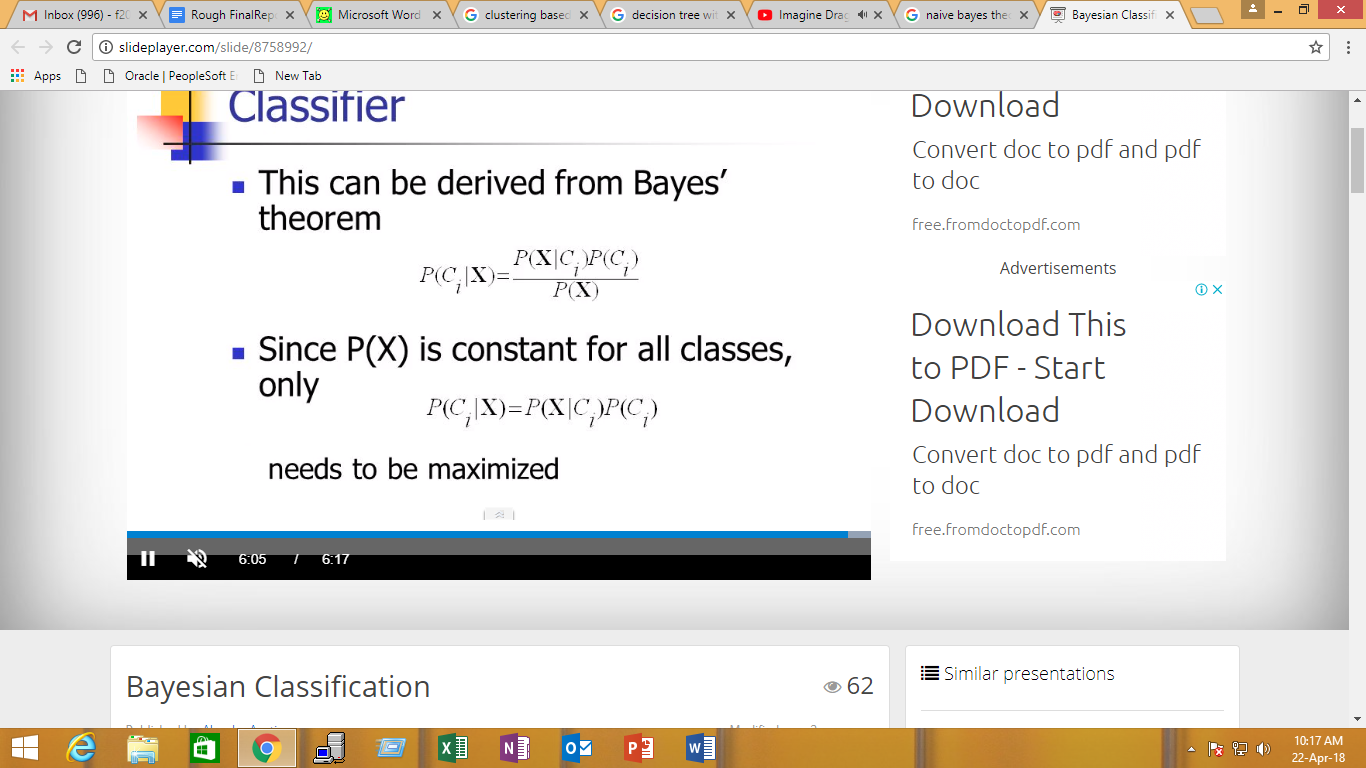
* Naïve Bayesian Classifier was implemented to generate probabilities for each of the key attributes and thereby predict the possible type of crime that could occur given the values of the 5 key parameters such as the day, time, month and area it occurred in.
* Decision Tree classifier with applied KNN filter was implemented to check
* DBSCAN algorithm was applied to formulate the distances between test data and available clusters determine whether the test data node belongs to any cluster and if so, which cluster; or whether the node is actually just noise.

The purpose of the classifiers is to predict the potential crime type in a specific location within a particular time in the future.

**2.2.1 *Naïve Bayesian Classification***

Naive Bayesian classifier is a statistical model of a group of algorithms based on Bayesian Theorem that predicts class membership probabilities using the equation given in Figure 2.3. While Bayes’ Theorem computes the probability of the occurrence of an event given the probability of another event that has already occurred, Naive Bayesian classifier assumes that every attribute being considered is independent of each other. Upon taking into consideration the independent nature of the attributes. the equation is changed as shown in Figure 2.3. Since the crime attributes chosen have an independent effect on each other, this classifier was an ideal choice. 

(i)



(ii)

FIGURE 2.3 i)BAYESIAN EQUATION ii)MODIFIED BAYESIAN EQUATION - NAIVE BAYESIAN EQUATION

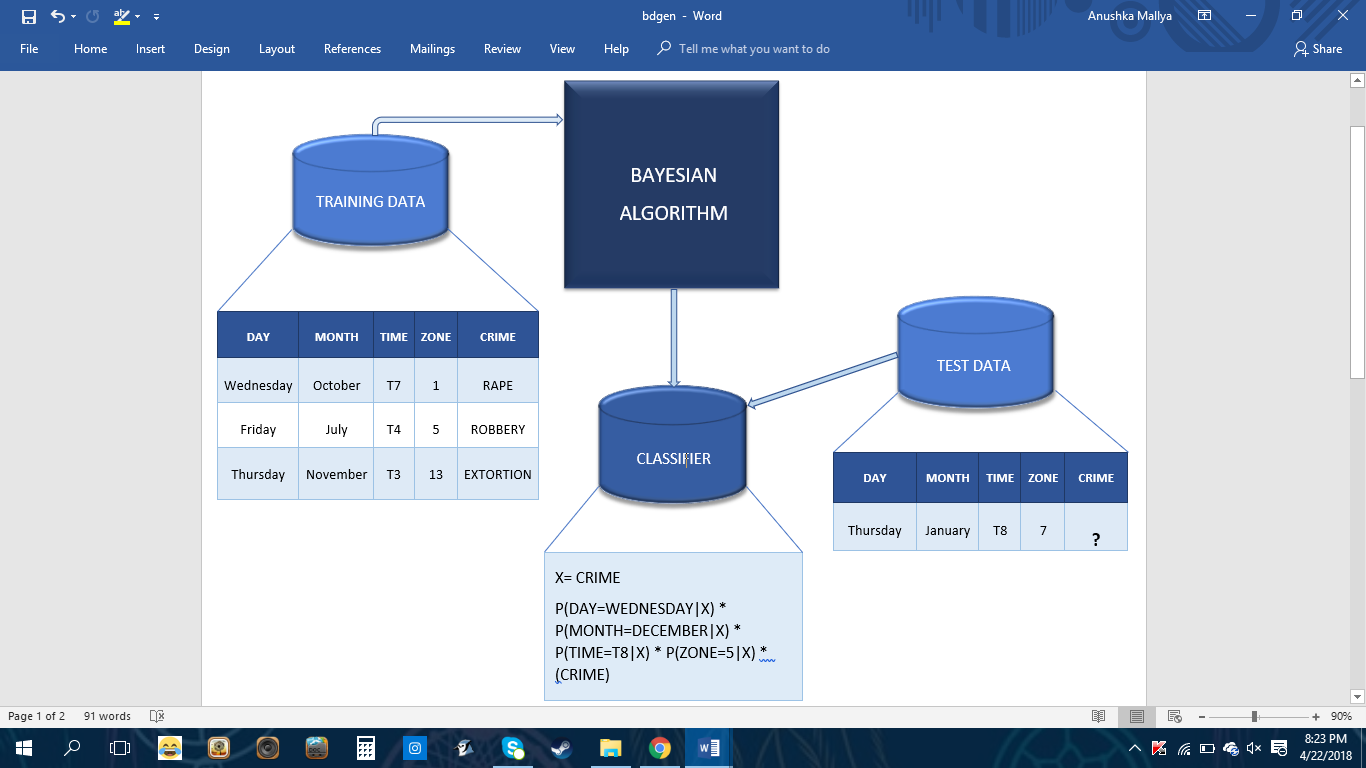


FIGURE 2.4. BLOCK DIAGRAM FOR NAIVE BAYESIAN CLASSIFIER

* The type of crime was selected as the class label and parameters such as day, month, time and area of the crime were considered to deduce which type of crime would be more likely to be committed.
* Through recursive scanning, the Bayesian algorithm generated matrices for each of the classes available in our training data set.
* A counter maintained for each of the 17 crimes was incremented to keep track of the number of crimes committed for that particular class.
* The Naive Bayesian Formula (as observed from Figure 2.4) was then applied by the classifier to calculate the likelihood of the occurrence of each crime based on the given test data input by the user.
* Upon obtaining the probability values for the various crimes being committed given the input data, we can deduce which crime is most likely to occur and which crime is likely to occur.

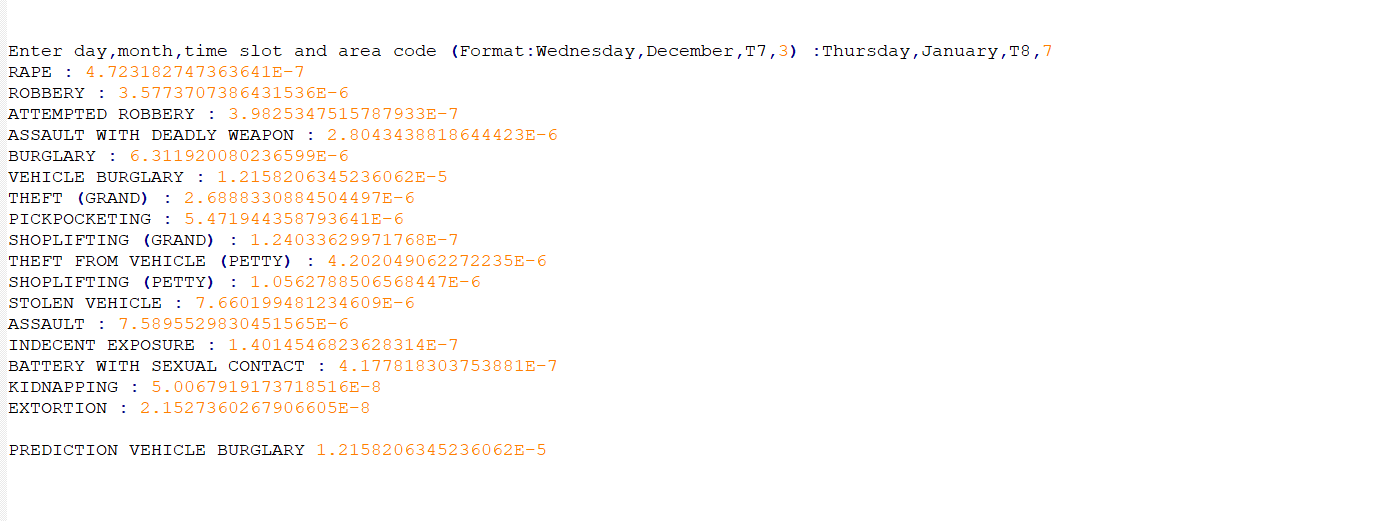
****

FIGURE 2.5. OUTPUT FOR TEST DATA (Thursday, January, T8, 7)

**2.2.2 *Decision Tree with KNN filter***

In Order to achieve higher accuracy we decided to implement a combination of two very unlikely classifiers one which create a model everytime as data is entered while other a lazy learner which implements the model when the input is requested to be tested. So we first see what each of them do.

Decision Tree :

A decision tree is a supervised learning algorithm that builds regression models in the form of a tree structure. It splits a dataset into smaller and smaller subsets and thus simultaneously develops associated decision trees incrementally. The final result is a tree with decision nodes and leaf nodes. The general purpose of using a Decision Tree is to create a training model which can be utilised to predict value or class of target variables by learning decision rules inferred from previously available training data.

KNN Filter :

KNN is lazy learning algorithm (i.e. it does not use available training data to establish any general rules ) and it is non-parametric (i.e. it determines the model structure based on available data and doesn’t make any assumptions based on the same.) Its motive is to use a database in which the data points are separated into several classes to predict the classification of a new sample point. Thus, the training phase involved is real fast.

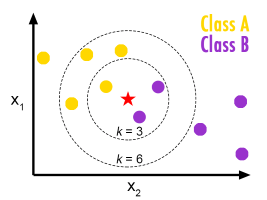


FIGURE 2.6. PICTORIAL REPRESENTATION OF KNN

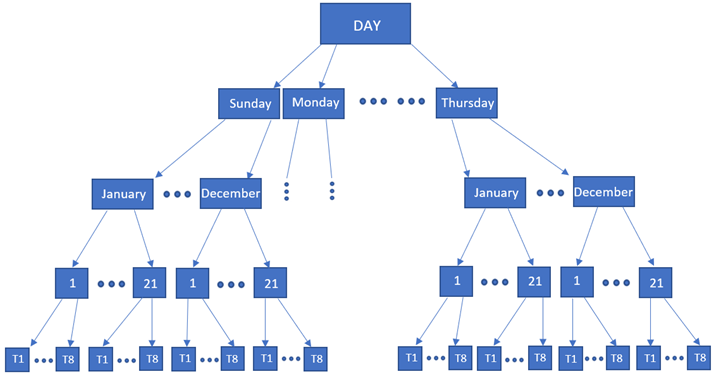
****

FIGURE 2.7. DECISION TREE FOR REDUCED CRIME DATA SET

* In our Decision Tree model (as shown in Figure 2.7), the root note Day has 7 decision nodes (e.g., Sunday) each of which have 12 more branches (e.g., January, February etc.) Each month branch is further divided into 21 area zones which are further divided into 8 time slots. These time slots happen to be the leaf nodes (e.g., T1, T2 etc.) which represents a classification or decision.
* The input is filtered similar to how Decision Tree would reduce the DataSet.
* For example our dataset of 148K entries is reduced to 148 / (7\*12\*21\*8) = 10 entries per input
* Upon reduction of the data set via the the Decision Tree classifier, the KNN classifier is applied to obtain k nearest values present in the Learning Set to our input data.
* The KNN Classifier is used to find the Euclidean distances between the remaining data and input data.
* This distance helps identify which neighborhood a particular test set value belongs to and
* Furthermore, the most frequent class determined by the k distances (in our case the top 5) are considered.
* Based on results obtained, a comparison between Actual and Predicted values of the type of crime committed can be made.
* As this led an accuracy of only 72% we improved on our model by modifying the Distance function used in the K nearest neighbour classifier. ( clearly demonstrated in results part ).
* We have used weighted distances meaning if the distance between our input and instance x is let's say “dist\_x” we multiply the inverse of “dist\_x” of that label with the “frequency” of that label. This implementation of the distance function helped us to get our accuracy from 72% to 84%. (output present in : [github\_output\_knn](https://github.com/MidasXIV/Spatial-Temporal-Analysis-of-Crime/blob/master/KNN/KNN_Output.txt) )

**2.2.3 *DBSCAN (Density Based Spatial Clustering of Applications with Noise) Based on concept of Dendrogram***

A cluster can simply be defined as a connected dense network of nodes which can grows in any direction often grouped by some property which makes a node present in one cluster more similar to another nodes in the same cluster and different too nodes in other clusters. Clustering is a good way to classify test data as data is grouped to a cluster in which there are similar elements. DBSCAN is largely used to find nonlinear shapes cluster based on density of node on a spatial plane. A node is mainly classified on property of density reachability and density connectivity.

Density Reachability can be defined using a simple example; Consider a point “x”, it is said to be density reachable from a point “y” if point “x” is within a predefined distance alpha from point “y” if point “y” has sufficient number of nodes in its neighbours which are also within the distance alpha.

Density Connectivity can be defined using a similar example let us consider 2 points “x” and “y” they are said to be connected if there exist a point “z” which has sufficient number of nodes in its neighbours and both the points “x” and “y” are within alpha distance. This is a recursive chaining process, so if “x” is a neighbour of “y”, “y” is a neighbour of “g” and “g” is a neighbour of “h” which in turn is neighbour of “Q” implies that “x” is neighbour of “Q”.

FIGURE 2.8. PICTORAL REPRESENATION OF HOW DBSCAN WORKS

In our project we have successfully implemented DBSCAN to visualize different crimes on the geographical map of California.Although DBSCAN is a very common clustering technique we were able to implement hierarchy as can be seen in the results part. By hierarchy we mean at each level of abstraction in our case the “zoom” level of the map we are able to assign the number of clusters in the first image as a very low zoom level we have about 20 clusters but on a closer zoom level we have multiple zoom levels and even clusters on further zoom. Based on the concept of clustering similar to dendrogram.

* We start of by defining the minimum distance alpha that a node must be in to be defined as density reachable to another node and then we define the minimum number of nodes that must be in the distance alpha from another node to be clustered.
* We iterate over all nodes to find the connected components of core points on the plane. Using this we can assign each node to a nearby cluster if the cluster is within alpha distance otherwise we assign the node as noise.

# Advantages of implementing DBSCAN

* Density-Based Spatial Clustering of Application with Noise Unlike other clustering algorithms do not require the number of clusters to be predefined.
* DBSCAN can easily point out outliers as outliers are points that lie alone in clusters that have low density these cluster or singleton item’s nearest neighbours are too far away.

# Although DBSCAN comes with fundamental pros it has few cons like :

* DBSCAN is not entirely reliable as some nodes that are reachable from elements in more than one clusters can be part of either cluster, depending on the order in which the nodes are present as order of the nodes influences the algorithm and thereby determine which cluster they belong to. A variation to this is to treat such nodes as noise, and this way we can achieve completely reliable clusters.
* The quality of the Clusters generated using DBSCAN depends largely on the minimum distance “alpha” used in the function to determine whether a node is Density reachable and connected to another node. The most common distance metric used is Euclidean distance. But this suffers greatly if the dataset used has high dimensionality called the “curse of dimensionality” which makes it hard to determine whether the distance measured is meaningful in relation to alpha.
* It is often hard to determine the minimum distance “alpha” as it requires proper understanding of the dataset in hand and the objective of the classifier.

**3. RESULTS**

**3.1 Naïve Bayesian Classifier**

We implemented Naive Bayesian Network classifier as our first classifier to see how well simple and naive classifier would tackle our enormous dataset. we implement Naive Bayesian Classifier with the aim to let the user input the day, month, time slot and area code; and the Bayesian algorithm would predict the crime that occurred for the inputted attributes. We immediately see how our dataset seems to be a little too complex for our classifier as the initial accuracy of the Naive Classifier reached to about 21%. Although the extremely low Accuracy achieved by our classifier it displayed all possible probability of crimes that can occur based on the given input ( can be seen in the images below ) this helped us realize the dimensionality of the dataset and how our dataset existed on closer examination to why such poor results we realize that based on the same inputs of Day month time and area there are many crimes occurring an average of about 12 crimes overlapping with our input data as a result the classifier as multiple “classes” on a single input and moreover we see how the data is biased as there are many occurence of common crimes like shoplifting and pickpocketing as compared to serious likes like “rape” and “assault” as a result our classifier rendered weak to the dataset so we implemented simple utility functions ( can be seen in code part under folder Utility Functions) to find out outliers and data which led to multiple class prediction. this led to the creation of Clusters based on DBSCAN we effectively remove all the outliers and increase the dimensionality of our data by simple tasks like 6 times slots to 8 times slots . 15 area codes to 21 area codes . this greatly boosted our classifier performance and it achieved a maximum of 78% accuracy on one of our datasets and 42% accuracy of the other. following are steps of implementation of this classifier from 21% accuracy to 78%. (Outputs and Visualization of the program can be seen on our github page.) (https://github.com/MidasXIV/Spatial-Temporal-Analysis-of-Crime/tree/master/BayesianNetwork)

The TRAINING DATA given as input looks something like this:



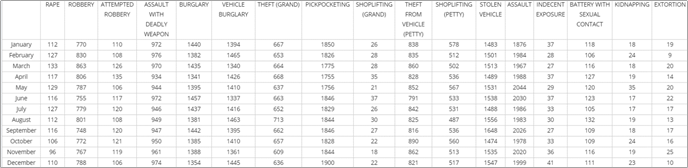
These are 10 entries from the set of about 148,000 entries.

**3.1.1 THE WORKING OF NAÏVE BAYESIAN CLASSIFIER**

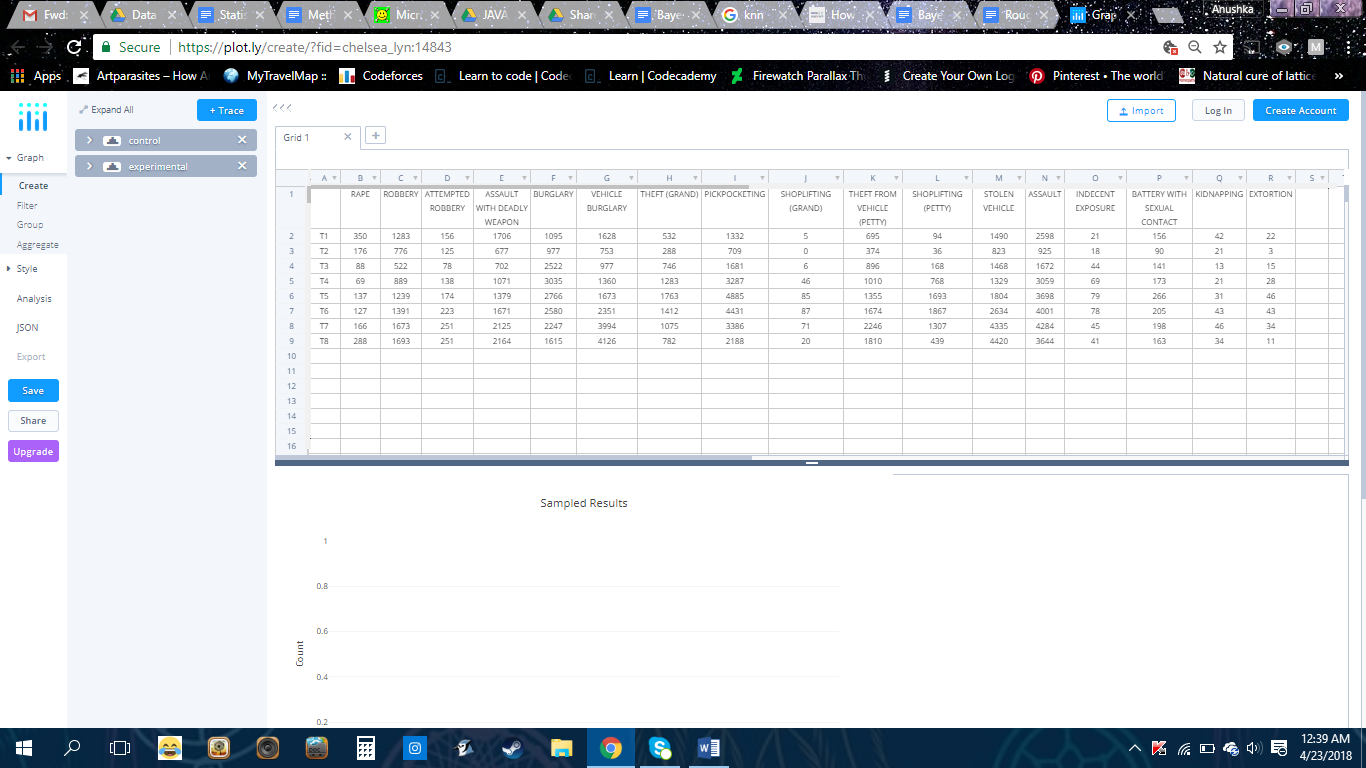
* The first step towards achieving this was to find the number of all the crimes that occurred for all the months of a year, all days of a week, all time slots and all the area codes. Below are the matrices created. This numbers corresponded to probabilities when divided by number of occurence of that crime in our dataset.

( Code written in BayesianNetwork.java )

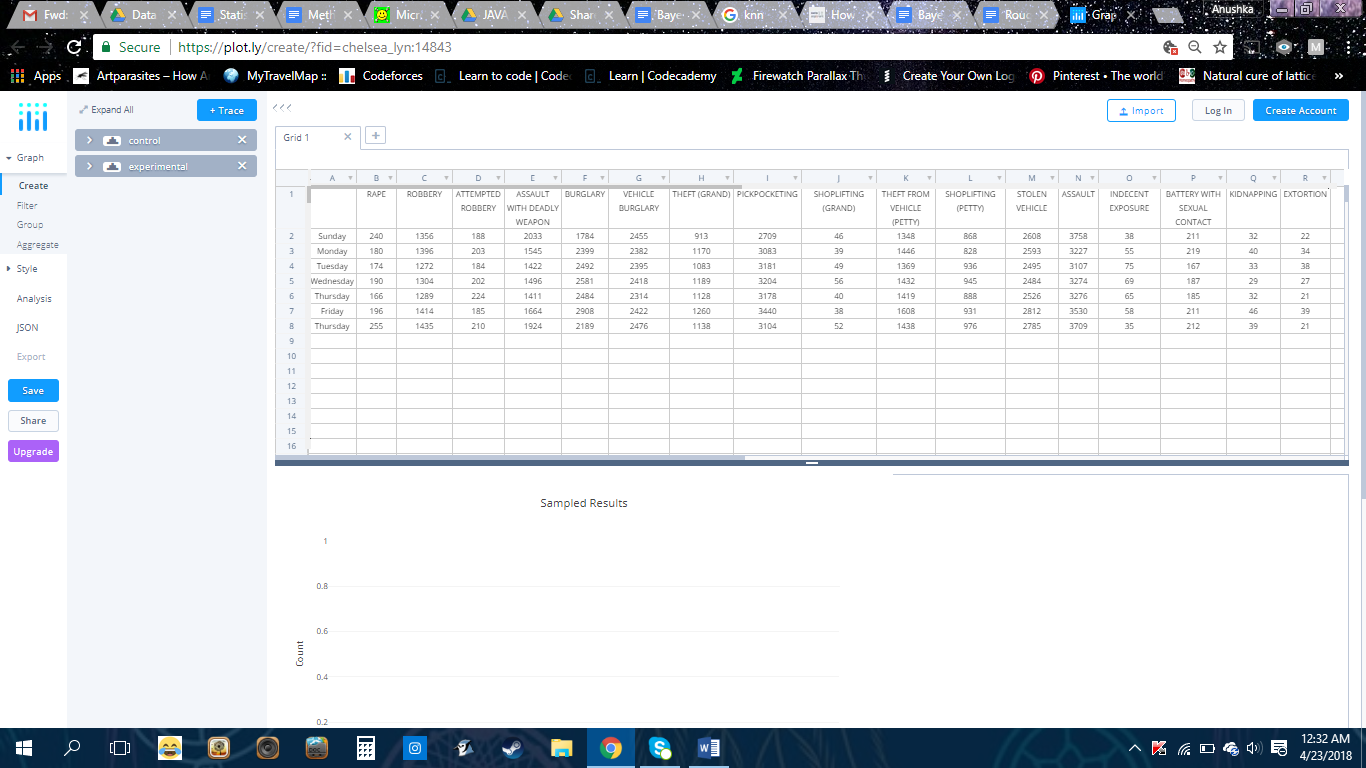
Month VS Crime



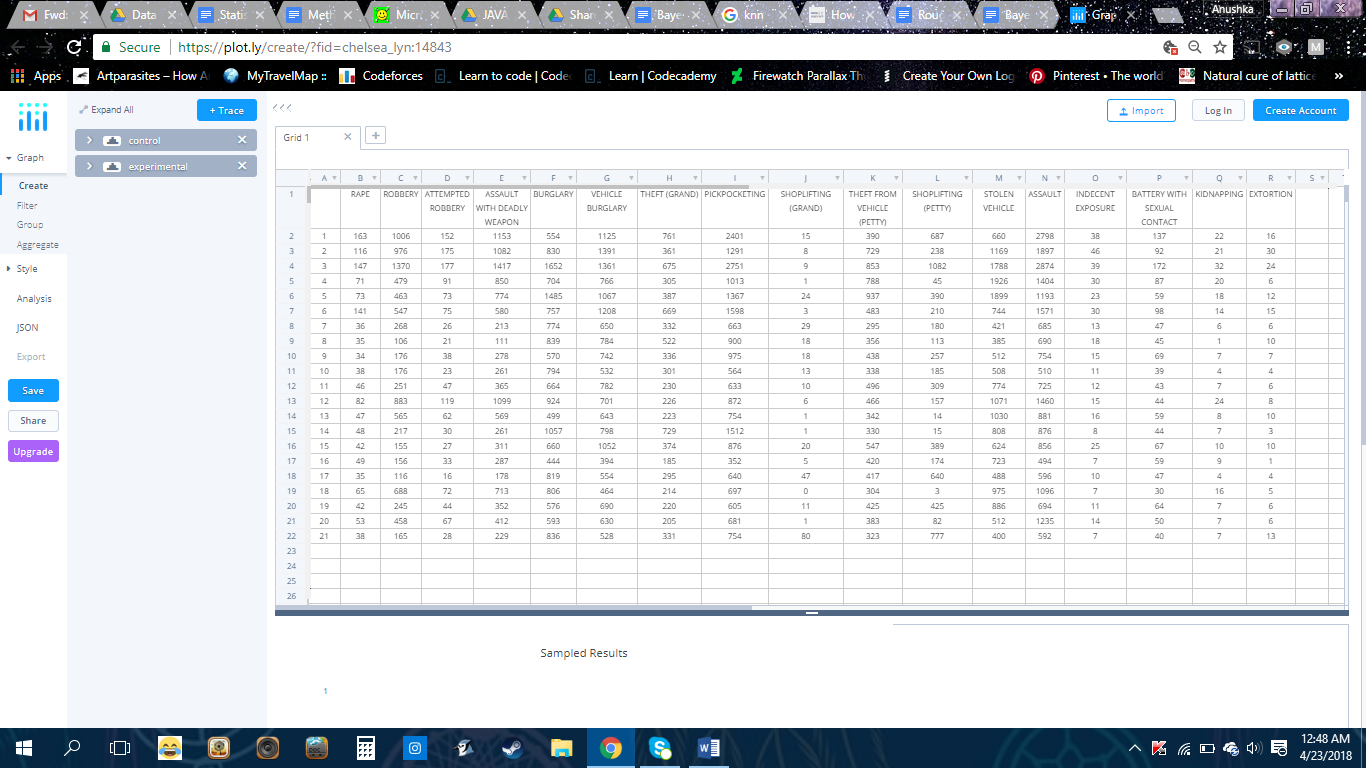
Day VS Crime



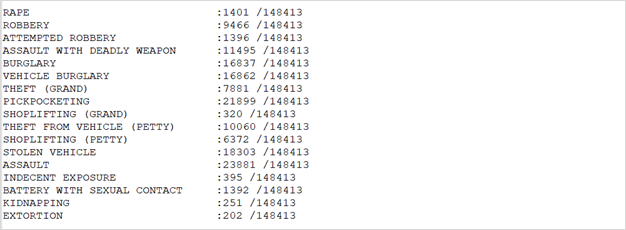
Time slot VS Crime



Area VS Crime



* The second step was to calculate the probabilities of each crime with respect to all the other crimes. each term divided by the number of occurence of that crime for example [January , Rape] is basically the probability that the month was january when the class if defined to be “rape” the table value gives us exactly this we just divide with by the number of occurence of that crime ( in this case crime = “rape”).

The output for this is as follows: 

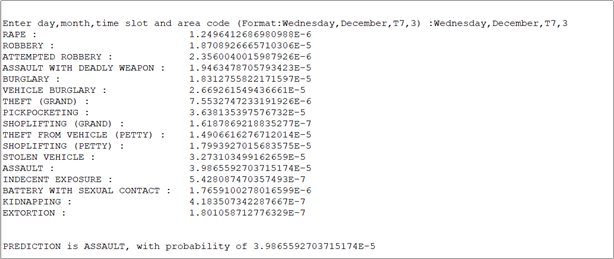
* The third step is the most important step. Here, we took the user’s input of the day, month, time slot and area code as the TEST SET and the predicted the Crime type for the given test set.  
  *For example, let’s say the user gives input of  
  Day: Wednesday  
  Month: December  
  Time zone: T7  
  Area code: 3*

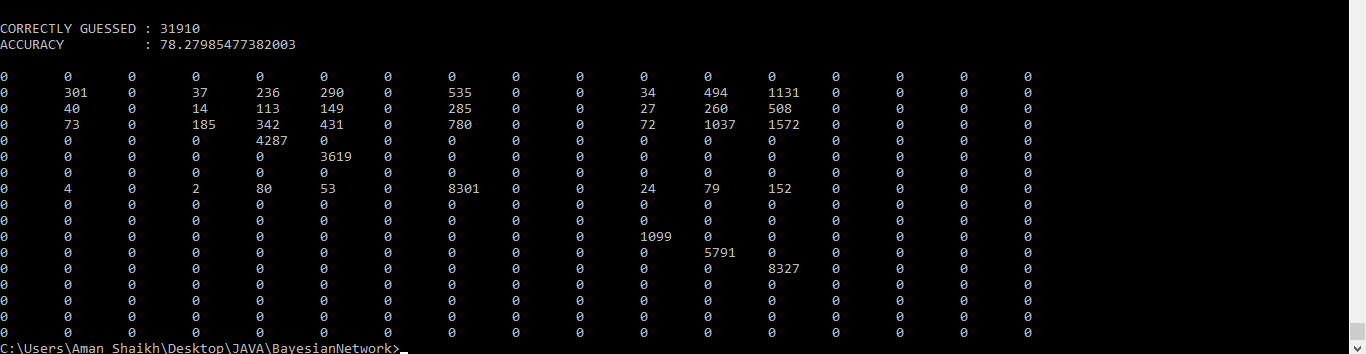
We calculate the probability of all the crimes given the test data and choose the crime with the highest probability as the predicted crime.

X= CRIME

P(DAY=WEDNESDAY|X) \* P(MONTH=DECEMBER|X) \* P(TIME=T7|X) \* P(AREA CODE=3|X) \* (CRIME)

*For our example, the prediction is assault as it has the highest probability. This data is then checked by the Verification Set. to optimise running time of the program we have combined the Test with the verification set so the program determines and keeps track of all its correct guesses and predictions.*

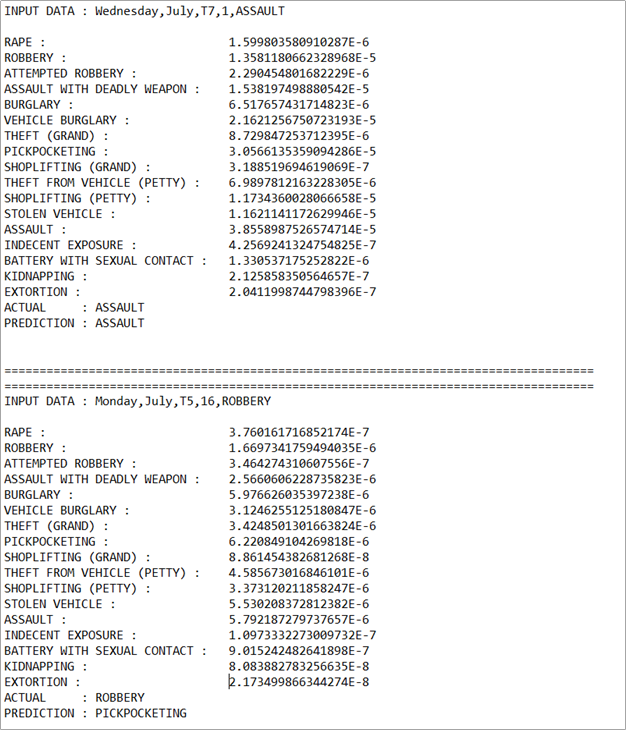




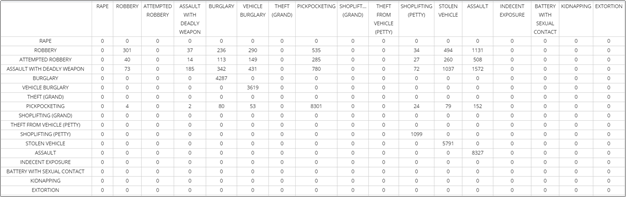
**3.1.2 Assessment : ROC Data**

ACTUAL DATA : We took a test sample which consisted of the day, month, time slot, area code and the corresponding actual crime type.  
PREDICTED CRIME/DATA : The test set sample is fed into the Bayesian Classifier algorithm and the crime with the highest probability is the predicted crime.

We took the test sample to be multiple variations of all the entries in the training dataset and predicted the crime type. The test set samples for which the predicted crime type matched the actual crime type, added to the accuracy of the ROC Model.

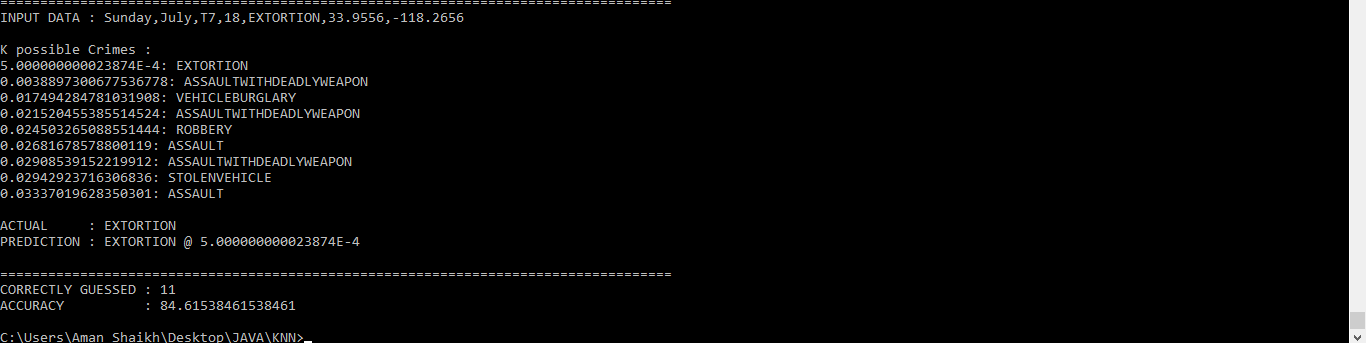
The final maximum accuracy was calculated to be 78%.  
Below are 2 test samples and their predicted values.   
The input of Wednesday,July,T7,1,ASSAULT gives the predicted crime type as assault, thus adding value to the accuracy.  
The input of Monday, July, T5, 16, ROBBERY gives the predicted crime type as pickpocketing, when the actual crime type is robbery. Thus, the prediction is wrong. 

The ROC Table:



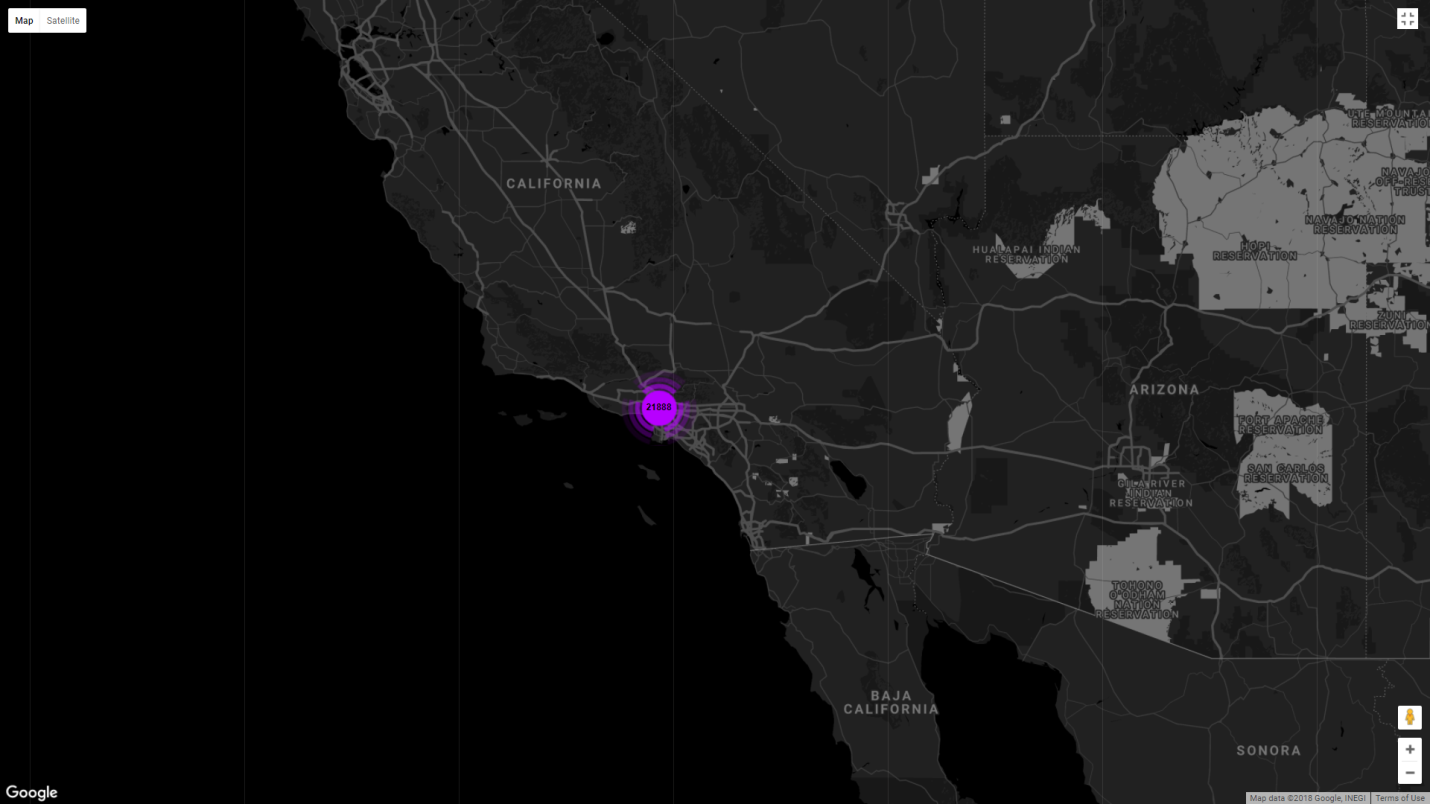
**3.2**  **Decision Tree with K Nearest Neighbour Classifier**

After seeing how the Naïve Bayesian Network Classifier performed under our current dataset we decide to implement a variation of Decision tree and K-Nearest Neighbour classifier. Our previous classifier worked on the inputs Days , Months, Area Code and Time Slots which gives us 7 \* 12 \* 21 \* 8 \* 17 possibilities even though this leading to be a massive number it wasn’t an ideal way to predict crimes but an excellent way to estimate all possible probability of each of the 17 crimes. A decision tree is a tree shaped structure which depicts a decision and every potential outcome of making that decision; So in our case we start our Decision Tree with the node “Days” which can have 7 possible outcomes after this each of those 7 children nodes are classified on the basis of month and those 7 \* 12 children nodes are divided on the basis of 21 area codes and then the 7 \* 12 \* 21 child nodes are further divided on basis of 8 time slots. Which means our decision tree is equivalent to 7 \* 12 \* 21 \* 8 \* 17 rules this would achieve nearly the same accuracy as that of the Bayesian Network so we decided to use two more inputs which are the Latitude and Longitude of the person in real time. Then we perform KNN on the output of the Decision Tree for example our dataset of 148000 is reduced to 148000 / (7 \* 12 \* 21 \* 8 \* 17) which gives us an approximate of 10 entries per query we then use the KNN classifier to perform Euclidean distance between the latitude and longitude of the Crimes present in our reduced Learnset given by Decision Tree and the latitude and longitude of the user we then sort them in ascending order along with their labels and find the majority of the minimum 5 distances ( k = 5) using this we have achieved an accuracy of nearly 78% to 84%.

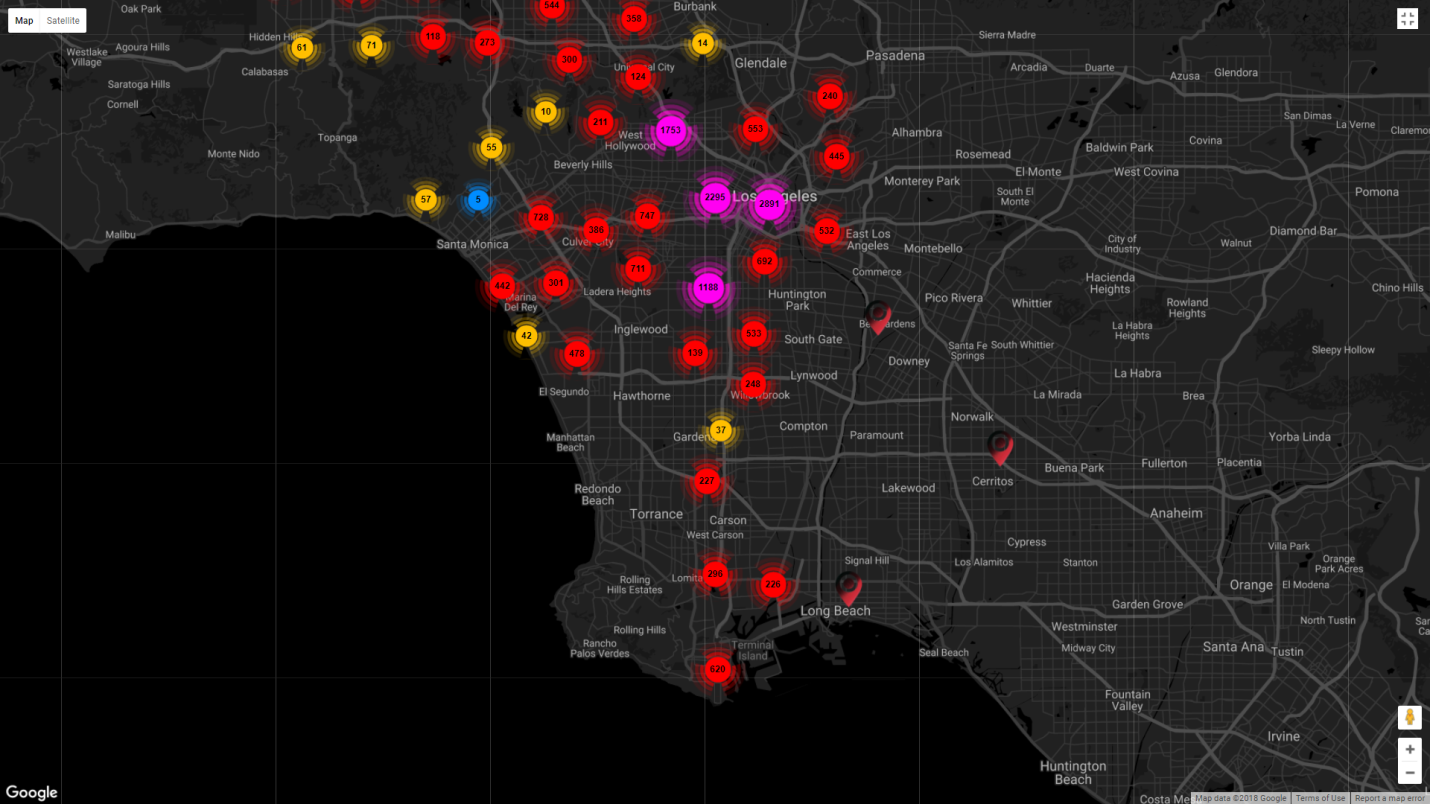
****

Here is an example of how the program work. The program displays the K possible crimes that can occur and displays their euclidean distance along with their labels. The K nearest neighbour classifier is slightly modified as you see in the image. we have used weighted distances meaning if the distance between our input and instance x is let's say “dist\_x” we multiply the inverse of “dist\_x” of that label with the “frequency” of that label. Like in the above example we can see that the distance between our input and Extortion is extremely low so inverse of this distance will make it extremely likely. similarly all the inverses are multiplied and then we check which has the highest possible chance hence our classifier selects Extortion rather than assault with deadly weapon. This implementation of the distance function helped us to get our accuracy from 78% to 84%. view detailed source code, input, output( <https://github.com/MidasXIV/Spatial-Temporal-Analysis-of-Crime/blob/master/KNN/KNN_Output.txt> ) and video of the program on (<https://github.com/MidasXIV/Spatial-Temporal-Analysis-of-Crime>)

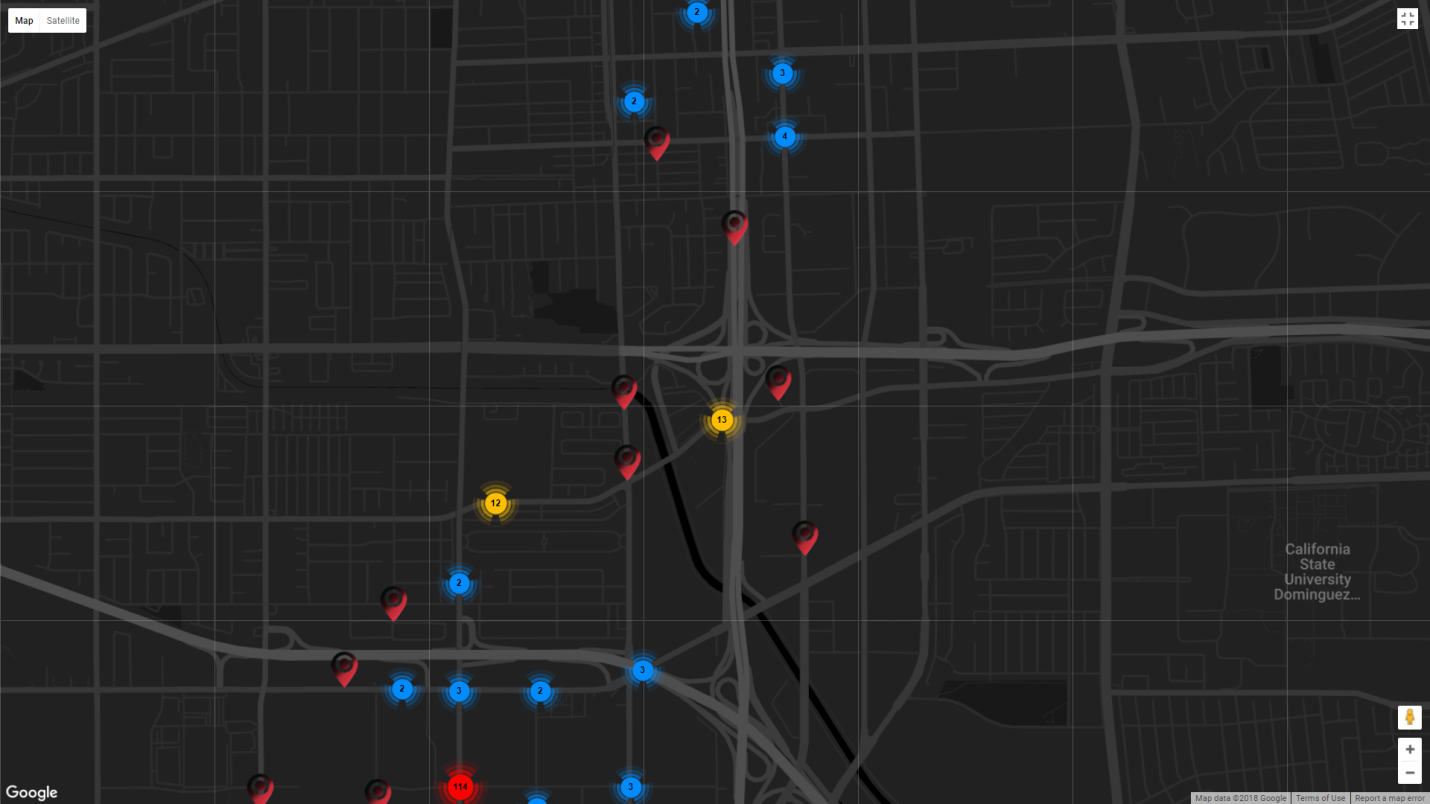
**Density based Clustering DB**

****

****

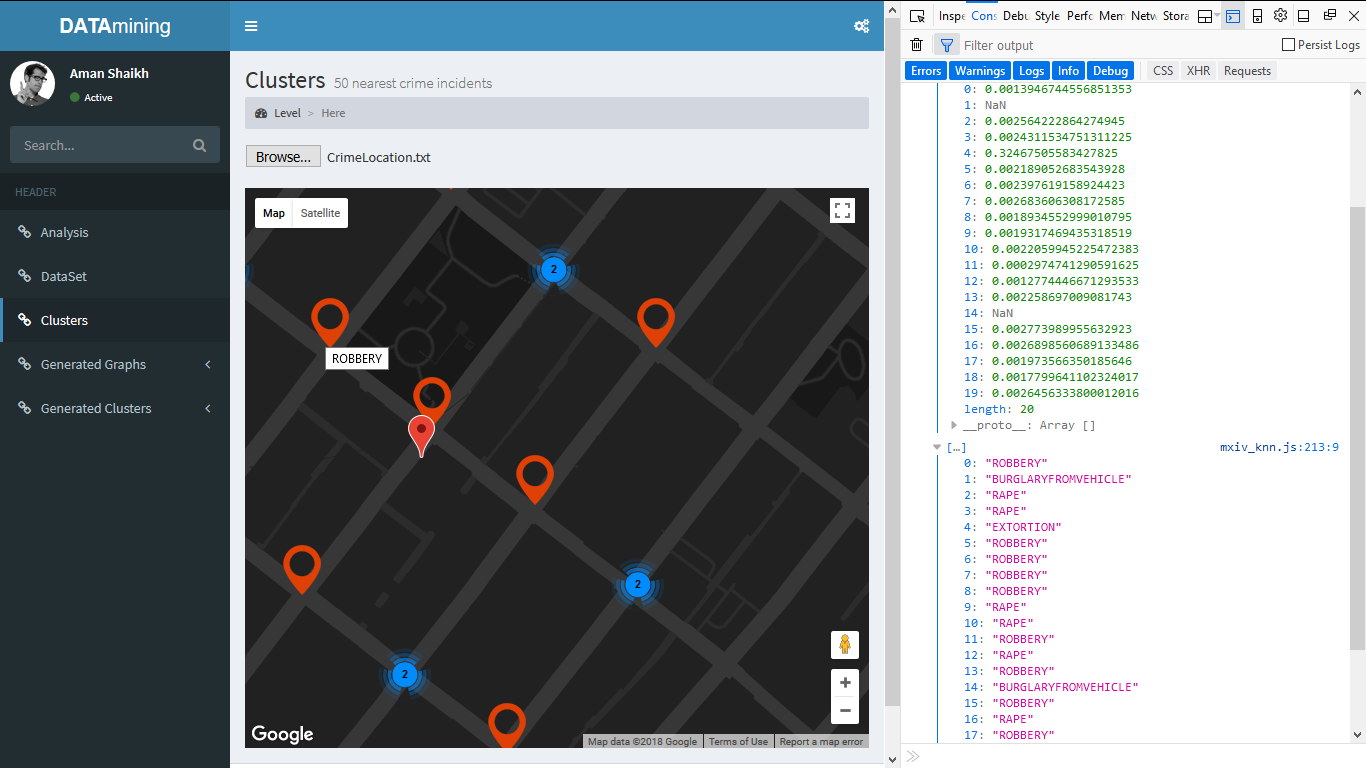
****

**SCAN**

****

As part of this project we were successfully able to implement clustering with hierarchical ability meaning the clusters were formed on the very primal and basic property of density meaning nodes belonging to a clusters are because they exist close to each other on the geographical plane as mentioned in “methodologies” under DBSCAN. Although DBSCAN is a very common clustering technique we were able to implement hierarchy as can be seen in the above two images at each level of abstraction in our case the “zoom” level of the map we are able to assign the number of clusters in the first image as a very low zoom level we have about 20 clusters but on a closer zoom level we have multiple zoom levels and even clusters on further zoom. Based on the concept of clustering similar to dendrogram.

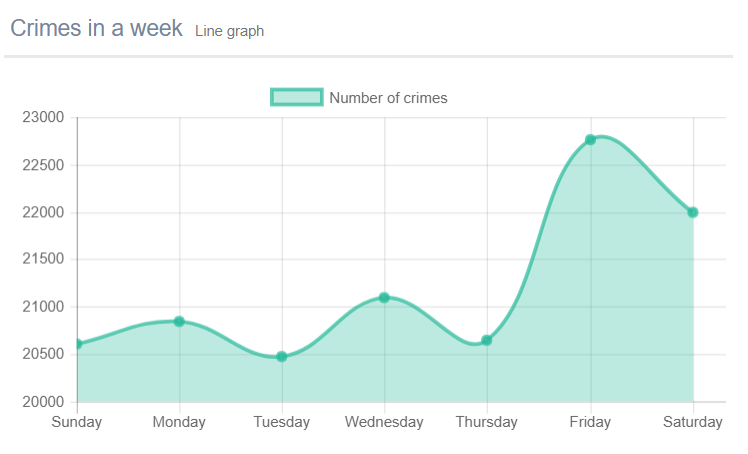
**Dynamic KNN Algorithm**

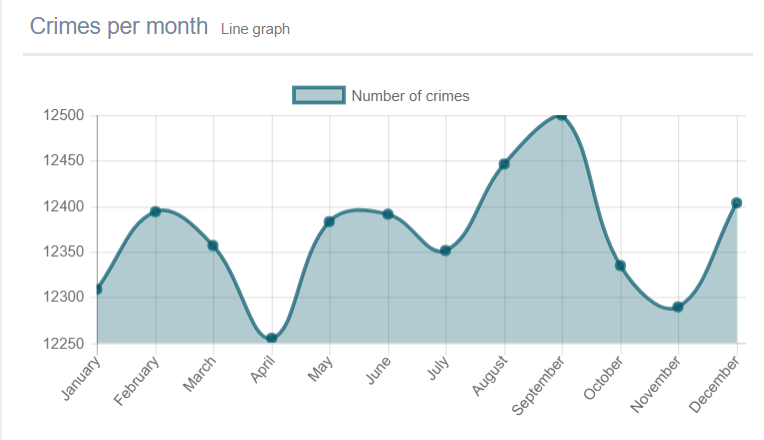
****

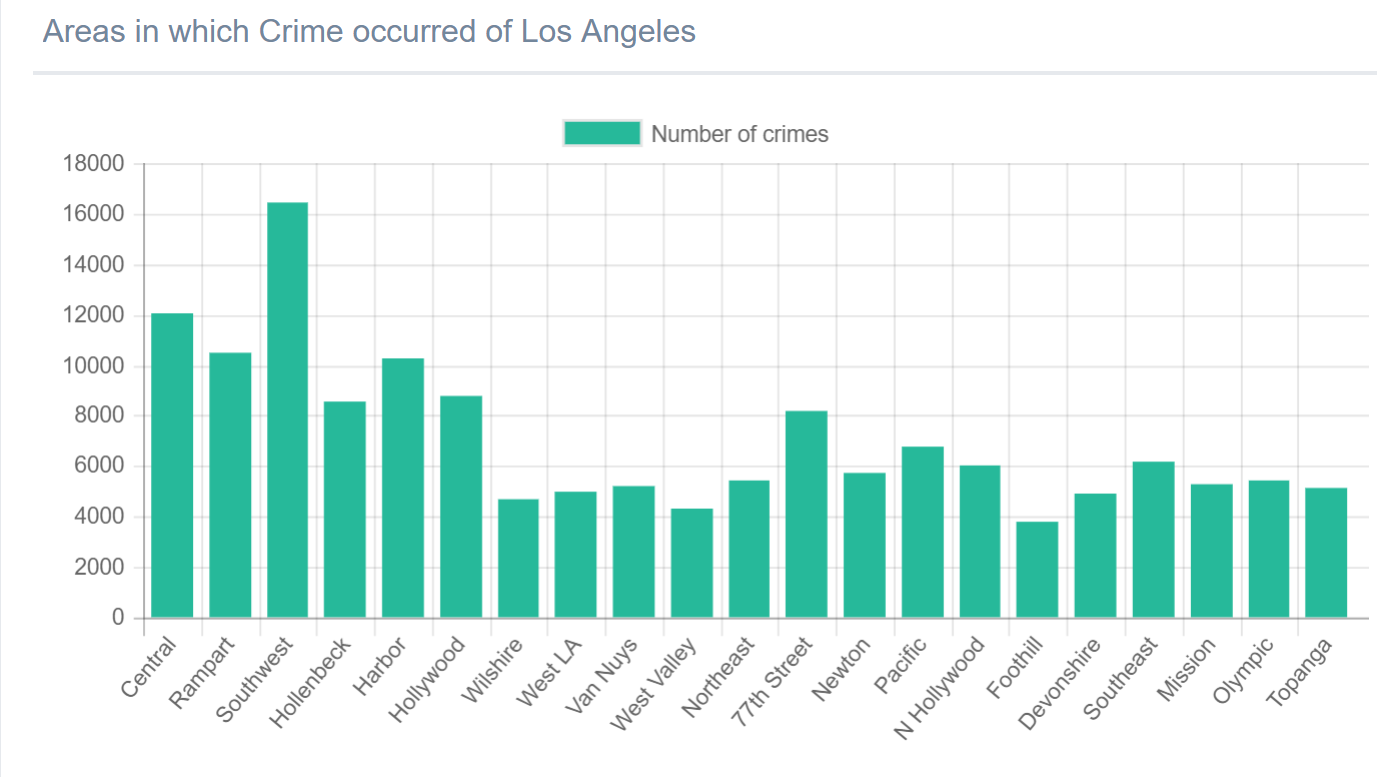
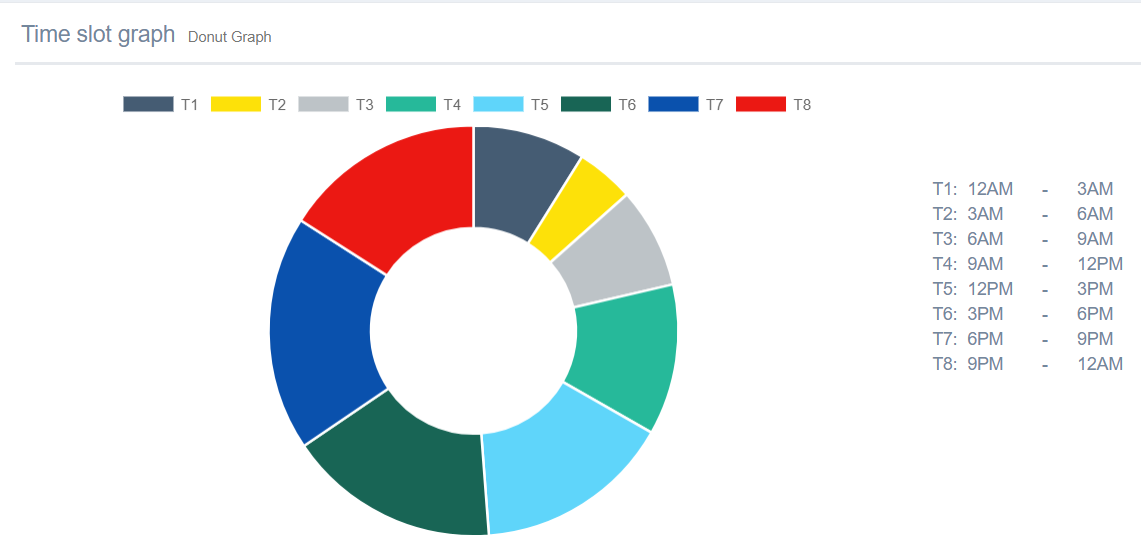
We took the above algorithm a step higher to allow users to interact with the algorithm by asking them to input data like day , month , time slot , area in which they want to check and then click on the google map of california which dynamically implements the above mentioned model to display the 20 nearest crimes incidents possible. You can notice the euclidean distances along with the possible crimes in the right sidebar. ( all visualizations demo can be found in our github ).

**3.3 GRAPHS**

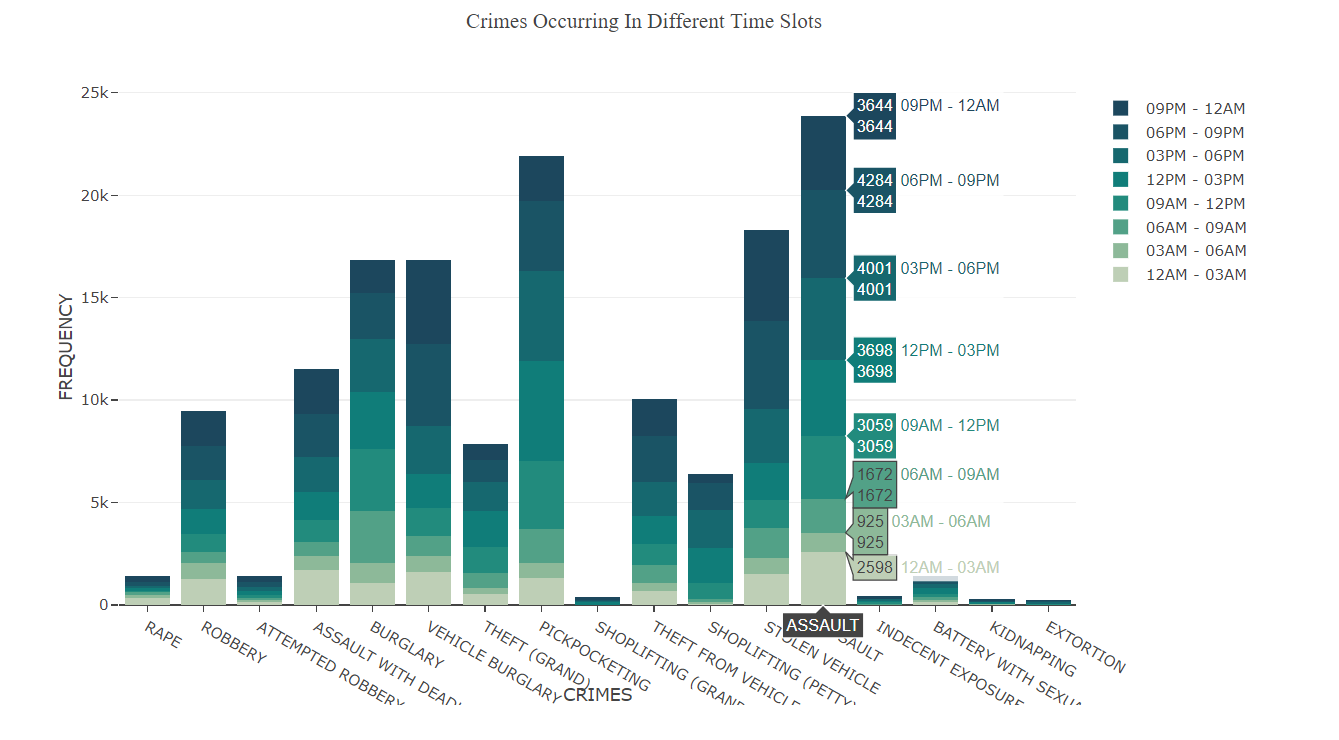
**3.3.1 Basic graphs**

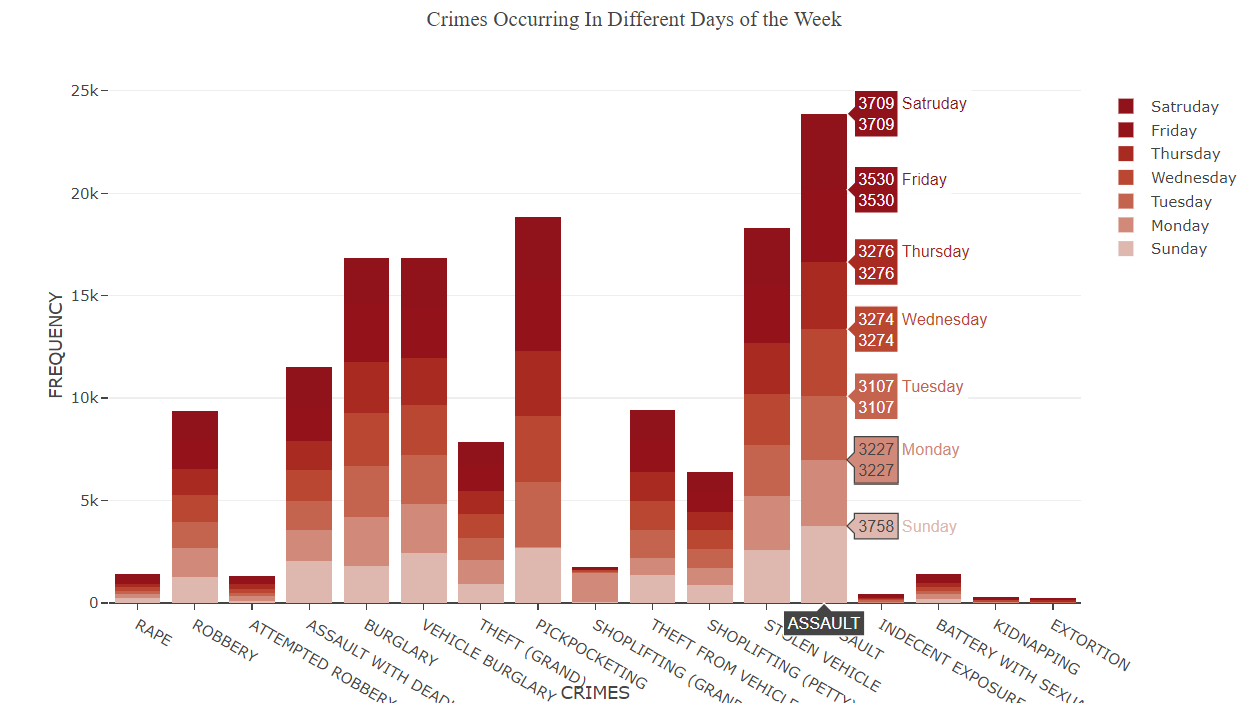
1. Crimes in a week graph:   
   This describes the statistical results for the total number of crimes for each day.  
   From this, we infer that the day when it is most likely that a crime will occur is Friday with 22762 crimes, while the safest day is Tuesday, with 20472 crimes.
2. Crimes per month graph:   
   This describes the statistical results for the total number of crimes for each month, in 6 years (2010 - 2015).  
   From this, we infer that the month where crime occurred was September, with 12,500 crimes. The safest month was April with 12,255 crimes. We see that there is not much gap between the safest and the most unsafe months, thus indicating that the number of crimes that occur, occur at a stable level throughout each month.



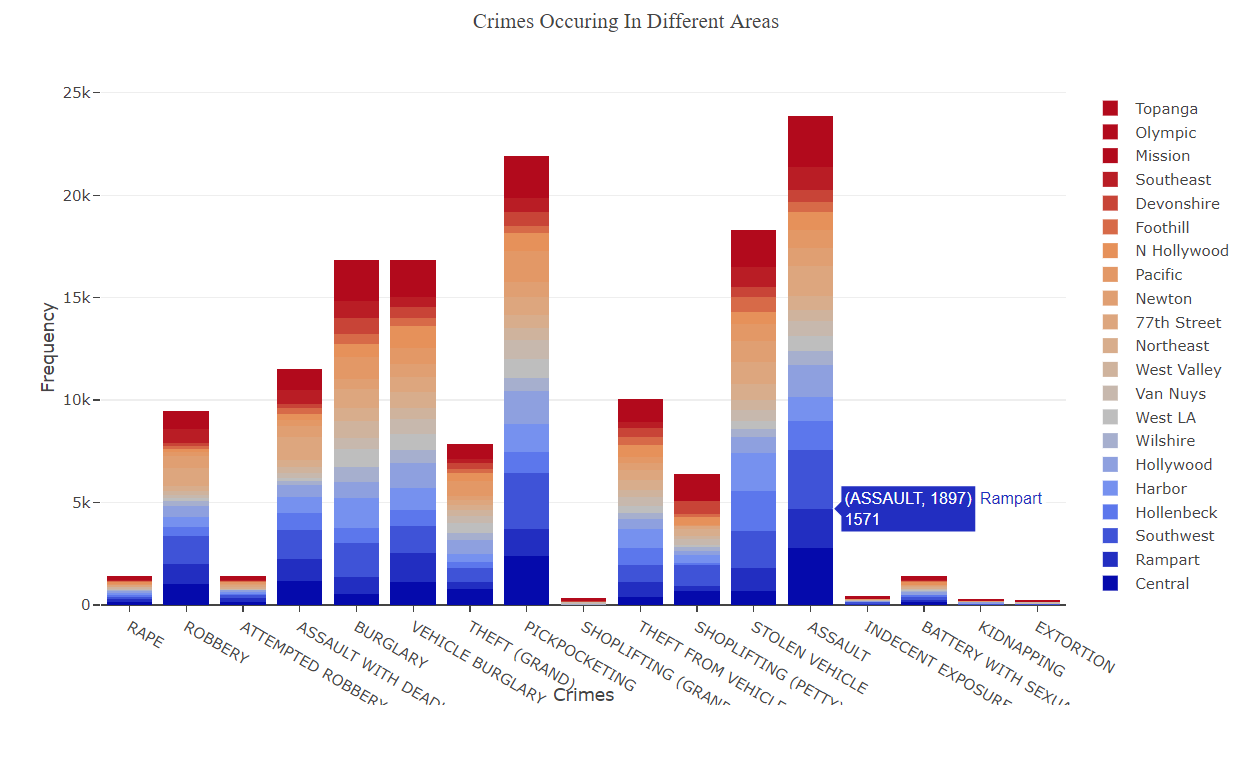
1. Crimes in the areas in Los Angeles:   
   This describes the statistical results for the total number of crimes for all the areas in Los Angeles.  
   From this, we infer that the area where it is most likely for crime to occur is SouthWest, which has a total of 16423 crimes, which the safest area is Foothill with a total of only 3792 crimes.
2. Crimes for every time slot:   
   This describes the statistical results for the total number of crimes for all the time slots, T1 to T8.  
   From this, we infer that the time slot T7 , between 6PM to 9PM, is the most highly dangerous time period of the day with 27483 crimes.   
   The time slot in which the least number of crimes have occurred is T2, which is from 3AM to 6AM, with 6771 crimes..

**3.3.2 Detailed graphs**

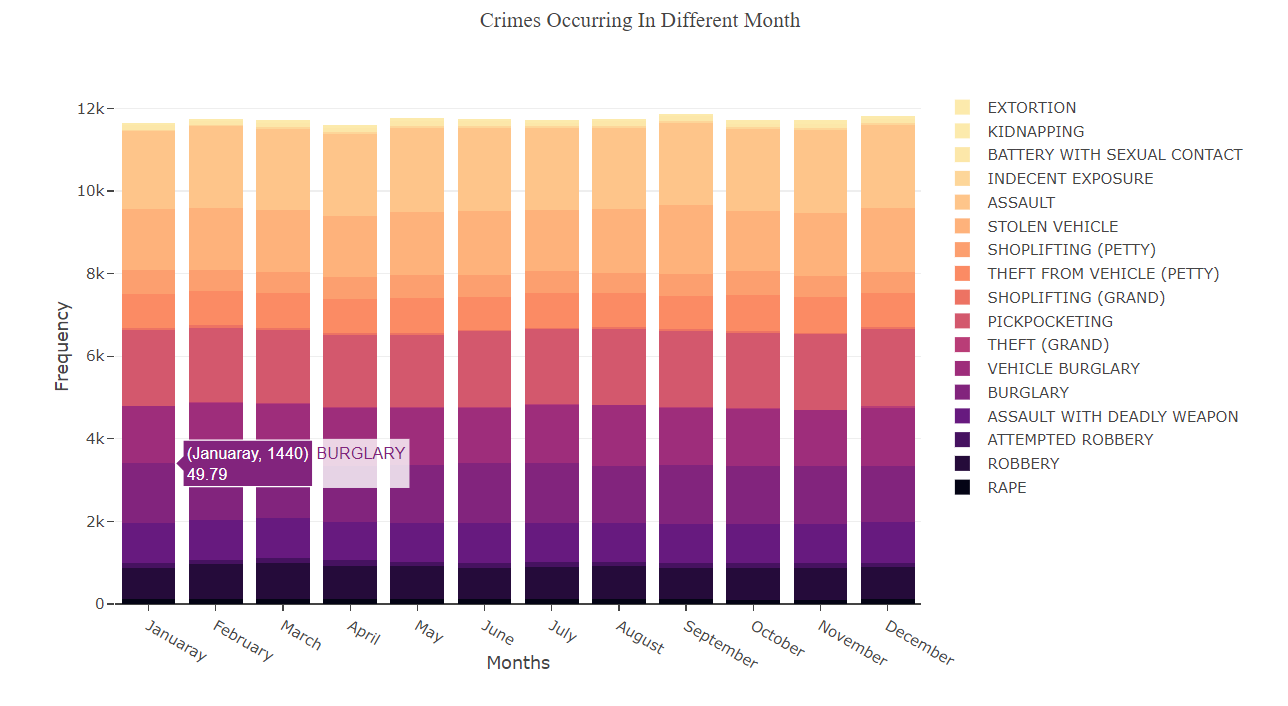
1. Crimes VS Time slots****

2. Crimes VS Days of the week ****

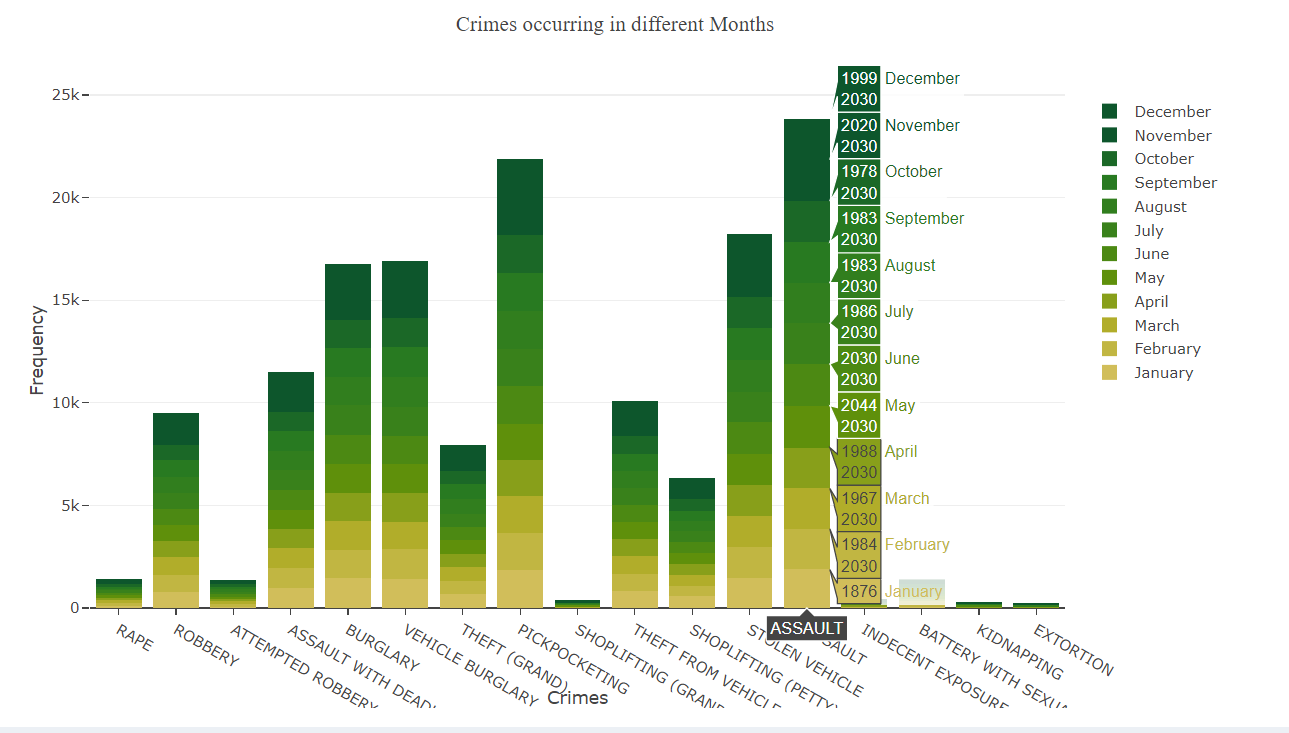
3. Crimes VS Areas of Los Angeles

****

**4. Months VS Crimes**

****

**5. Crimes VS Months**

****

In this paper we perform Tactical Crime analysis which involves analyses of real-world crimes dataset of Los Angeles in California from 2010 to 2016 and generate variety of scatter graphs in order to assist officers and investigators in identifying and understating current patterns of criminal activity. Our study aims to find spatial and temporal criminal hotspots in Los Angeles using a set of real-world datasets of crimes. In order to extract frequent patterns of crime on the dataset we use several technique and algorithms to generate a variety of graphs to give us better understanding of the data and also be able predict the crime rates in specific locations at particular time. We also intend to provide statistical analysis of different crime types with its demographic information.

Our dataset will include 6 distinct crime types including Assault, Drugs Alcohol, Burglary, Vandalism etc. The crime month, crime day, crime time, crime neighbourhood along with the latitude and longitude values to help generate a scatter graph of crime type versus the day/month/time.

The first goal of our study is to find spatial and temporal criminal hotspots. We aim to examine the results of Apriori algorithm and K-means cluster analysis technique on the Los Angeles dataset, then choose the model that gives the best results.

In the Apriori algorithm frequent item sets ordered by the crime location, day the crime was committed, the month, and the time period were provided. We are able to list out the most likely crime locations along with their frequent occurrence day and time.

The k-means cluster Analysis technique was combined with Geographic Information System (GIS) for detecting crime hot spots to deﬁne a spot as ”hot” or not in crime analysis.

The second target for our study was to predict the crime type that might occur in a specific location at a particular time period. The decision tree classifier generated via the ID3 algorithm enables us to achieve this target. To predict an expected crime type or crime rate, you need to provide four related features of the crime. The required features are: crime location namely the neighbourhood or area, the date of occurrence from which we can deduce the day of the week, the month and the occurrence time period.

After accomplishing our main goal by locating spatial and temporal criminal hotspots and predicting potential crime types, we applied some demographics analysis using Los Angeles neighbourhood demographics dataset to identify dangerous neighbourhoods in LA.

Crime rates play a crucial role while determining whether or not one should move to a new city or which neighbourhoods are to be avoided when they travel etc. Though crimes could occur at anytime, anywhere, it is frequently noticed that that criminals act on opportunities they find in areas most familiar to them. By exploring the Data Mining approach to determine the most criminal hotspots and find the type, location and time of committed crimes, we hope to raise public awareness about regions that are to be avoided, especially during particular timings. Furthermore, we seek to enable people to make more sound decisions with respect to their accommodation. On the other hand, police forces can utilise the acquired information to better allocate police resources and predict and prevent crimes from taking place more effectively. By ensuring all of this information is available at the disposal of the concerned authorities, we aspire to make our community safer for all citizens and tourists who would be visiting.

**Literature Survey**

1. Tahani Almanie, Rsha Mirza and Elizabeth Lor(2015) focused on finding the temporal and criminal hotspots. They compared crime datasets by statistical analysis. They used Apriori algorithm to produce patterns for criminal hotspots. For predicting the crime types, they made use of Decision Tree classifier and Naïve Bayesian classifier. It also uses demographics of the areas in order to study the factors of safety in the area.

2. (Name) (2016) researched about the procedure involved and objectives of various data mining techniques that could be employed while analyzing crimes and identified Association Rule Mining, Clustering (which includes K-Means algorithm, Hierarchial Clustering, Expecatation Maximization) and Classification methods (which includes Decision Tree, Nearest Neighbor and Neural Networks) as suitable techniques.

3. Gourav Govindaswamy, Vinod Kumar Kethineni and Santhosh Kumar P (2017) deduced that Partition Clustering is the most appropriate technique for drawing patterns and finding similarity measures and that Neural Networks such as “Blue Brain” have the remarkable ability to derive meaning from complicated and imprecise data which would prove extremely vital to extracting patterns and detecting trends that are too complex to be noticed by humans or other computerized techniques.

4. Mehmet Sevri, Hacer Karacan and M.Ali Akcayol (2017) researched on Association Rules using Apriori Algorithm. The Apriori Algorithm helped define association rules which further helped establish relationships between the attributes of different criminal records thereby making it easier to predict the unknown parameter of a particular case such as weapon used, victim profile etc.

5. Zakaria Suliman Zubi and Ayman Altaher Mahmmud (2014) explored the K-Means algorithm which takes into consideration input-Crime type, number of clusters, number of iterations and seeks the derivation of information from a ‘confusion matrix’.

6. In the journal "Crime Data Analysis Using Data Mining Techniques to Improve Crimes Prevention" (2014), DR. Zakaria Suliman Zubi describes different approaches to crime analysis such as Tactical, strategic, Administrative, Investigative crime analysis and the analysis of data such as Intelligence, operations analysis and how each of these help the crime analysts to develop strategies to combat crime. It’s one of the first papers which tries to find a pattern by performing Apriori Algorithm to attributes such as Crime type, gender age and marital status.

7. Dr. S. Santosh Baboo (2015) published a paper to help predict the crimes patterns and fast up the process of solving crime using clustering algorithms. This was one of the first journals that aimed to help crime analysts in India as it was inspired by the 26/11 Mumbai attack, the proposed solution was 89% accurate in predicting the general crime rate and was about 4% more accurate than previous algorithms used.