**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.

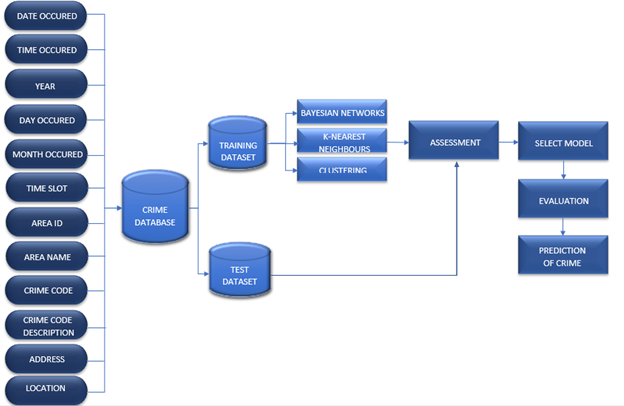


FIGURE 2.1. PROCESS INVOLVED IN PREDICTION PROCESS

**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.

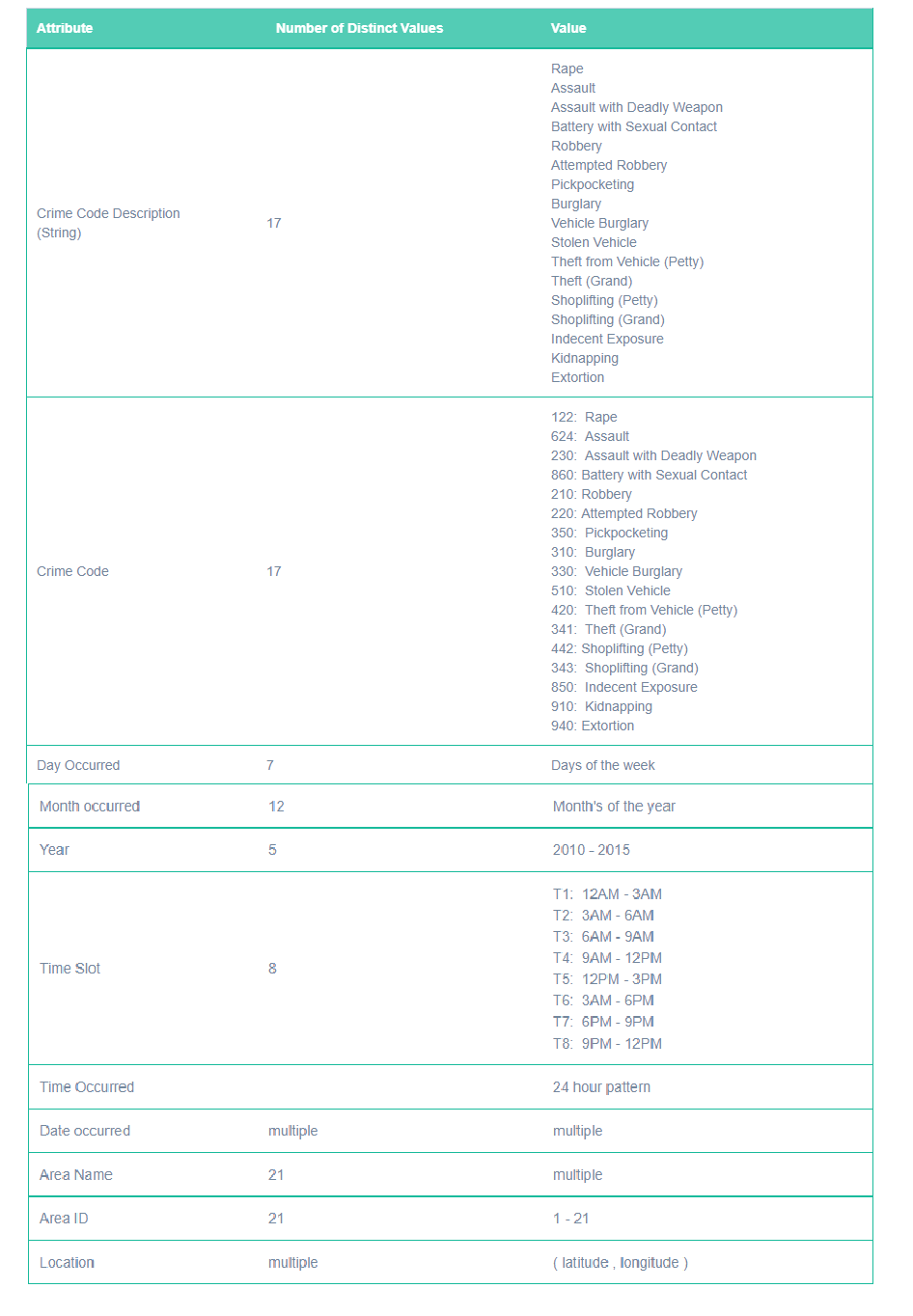


FIGURE 2.2. DATA SET ATTRIBUTES TABLE

**2.2 Data Analysis**

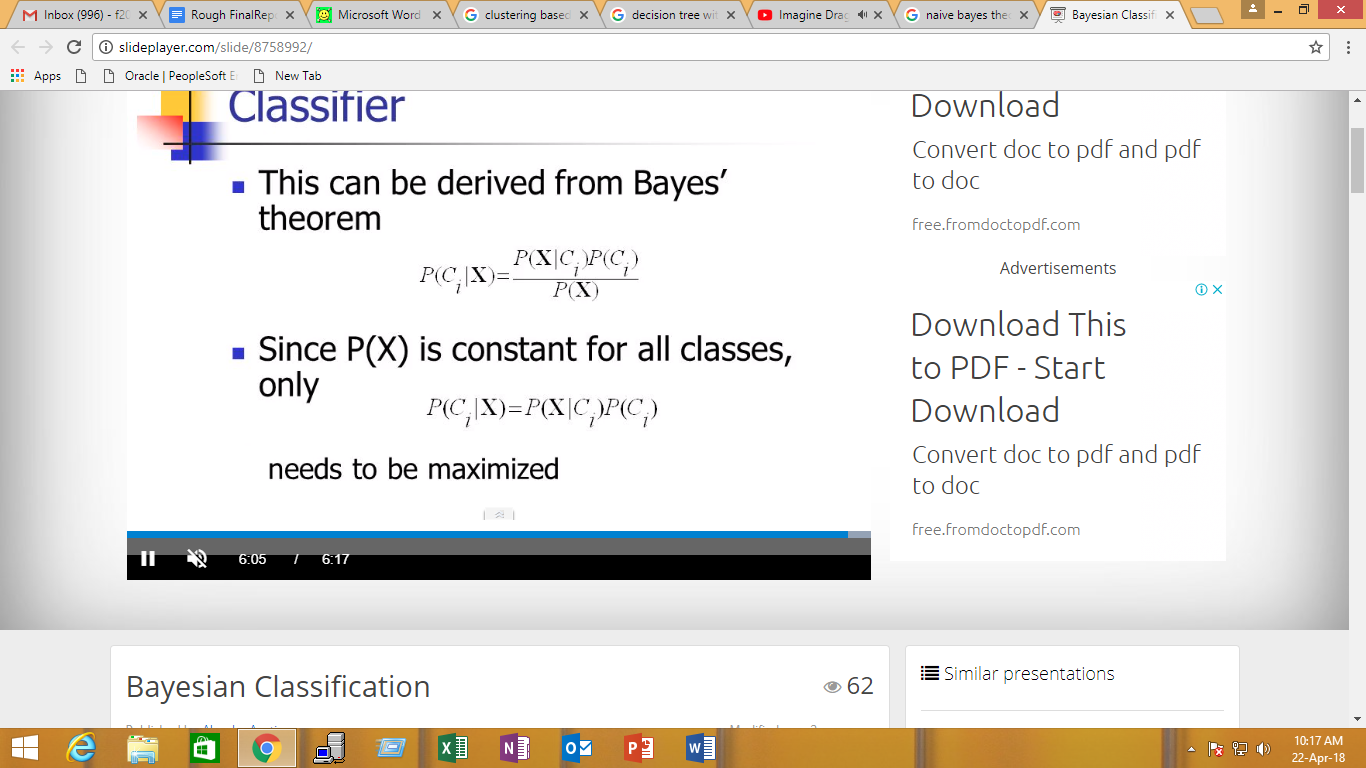
In order to deduce conclusive patterns from the crimes that were committed in Los Angeles, Naive Bayesian classifier, Decision Tree with KNN filter applied and Density based Clustering algorithms were applied on the available data set.

* 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.

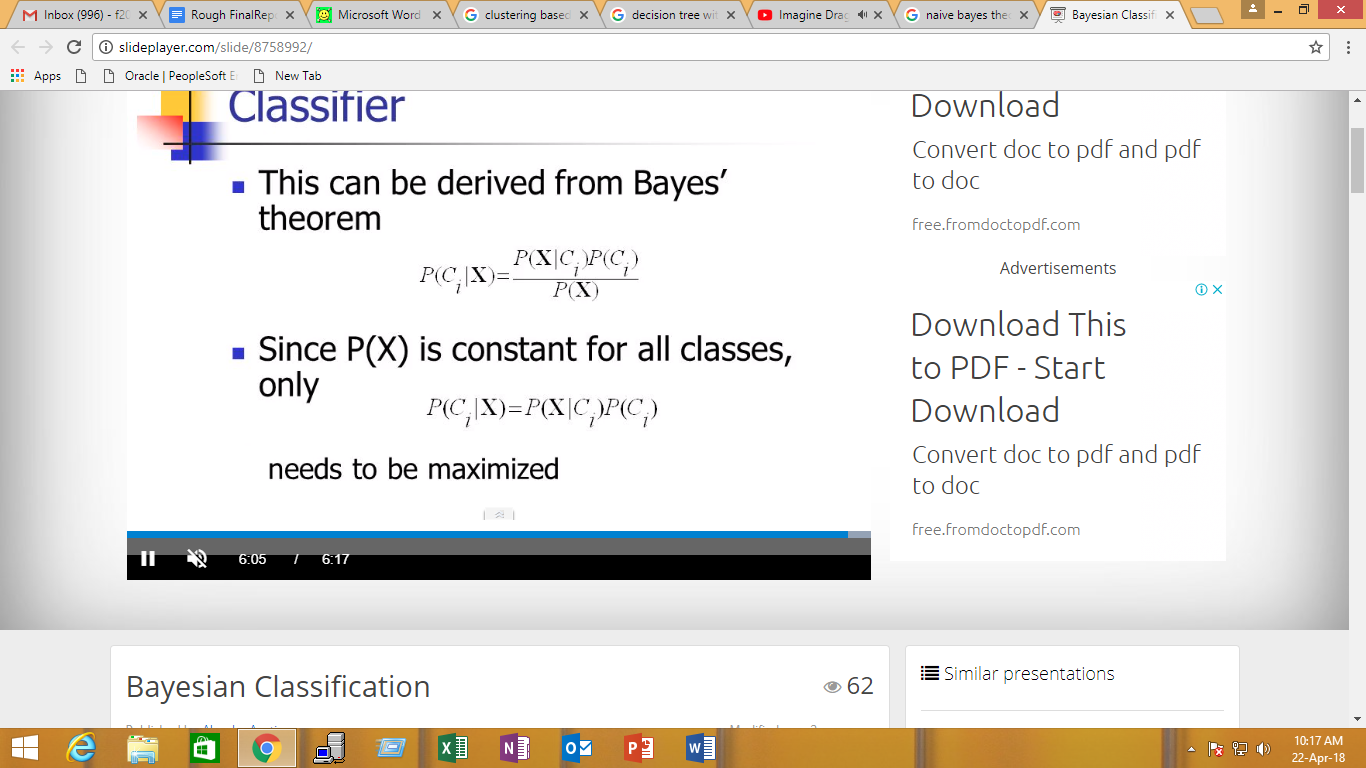
The purpose of the classifiers is to predict the potential crime type in a specific location within a particular time in the future.

**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

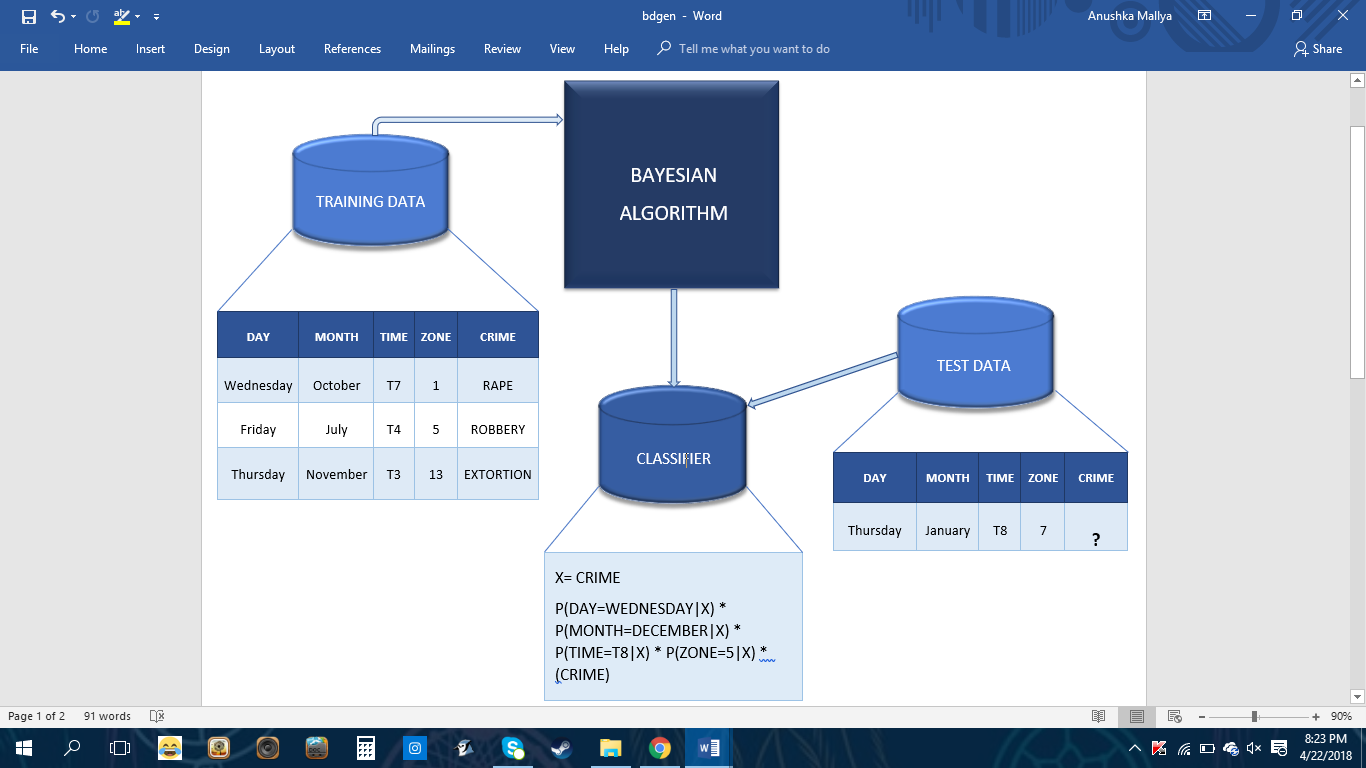


FIGURE 2.4. BLOCK DIAGRAM FOR NAIVE BAYESIAN CLASSIFIER

* 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.

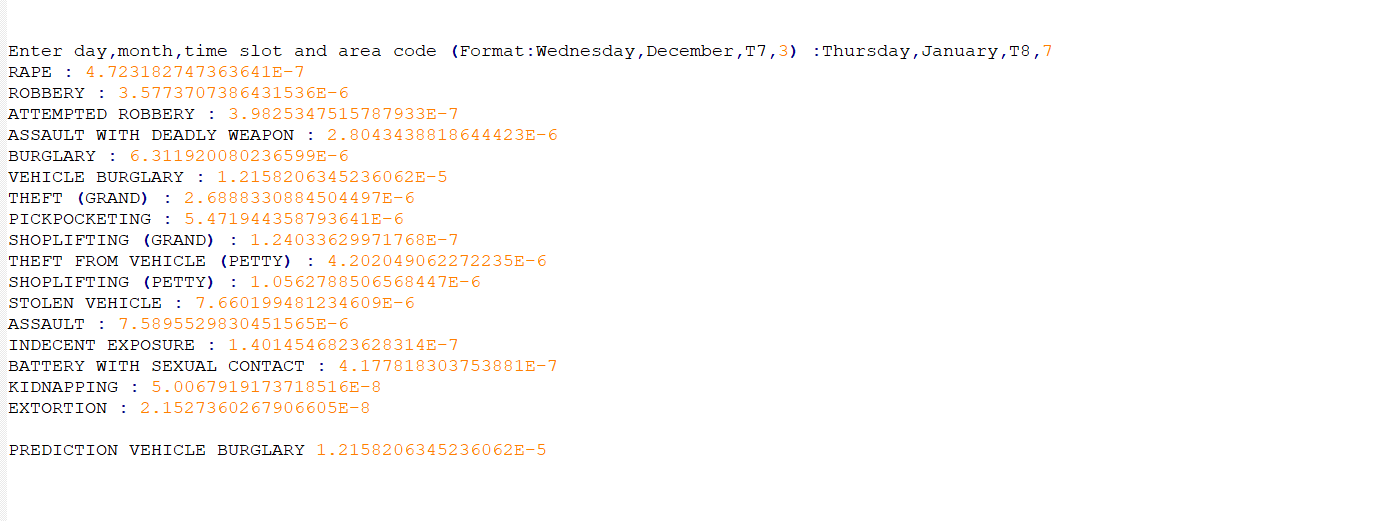
****

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 every time 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.

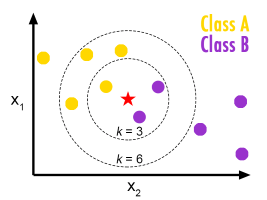


FIGURE 2.6. PICTORIAL REPRESENTATION OF KNN

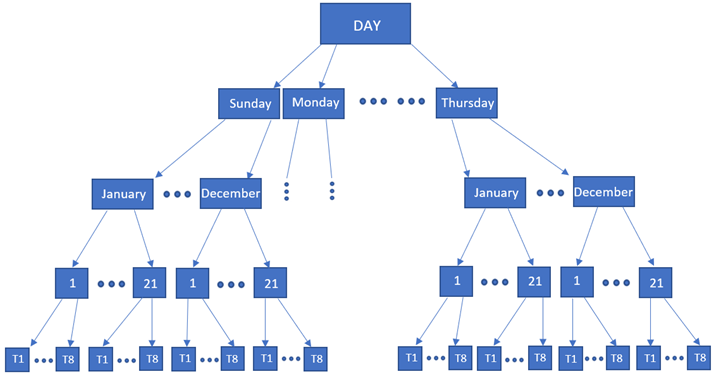
****

FIGURE 2.7. DECISION TREE FOR REDUCED CRIME DATA SET

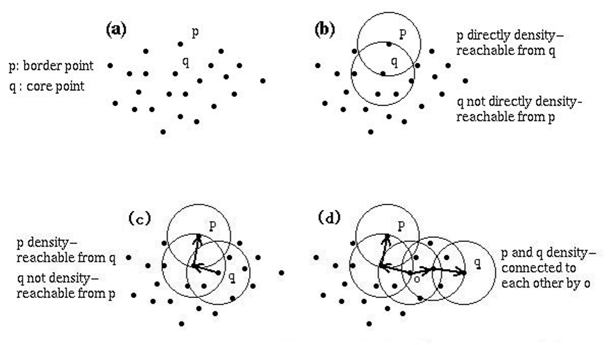
* In our Decision Tree model (as shown in Figure 2.7), the root note Day has 7 decision nodes (e.g., Sunday) each of which have 12 more branches (e.g., January, February etc.) Each month branch is further divided into 21 area zones which are further divided into 8 time slots. These time slots happen 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 Data Set.
* 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.