CS 4202 : FINAL YEAR PROJECT REPORT

Inquisitor : Crime Data Analytic Platform

by

G.H.D.M.M. Gammanpila (120165D)

A.M.M. Gangananda (120168N)

B.S. Kalansuriya (120282H)

W.D.T. Piyadasun (120478N)

Supervisor:

Dr. H.M.N. Dilum Bandara

THIS REPORT IS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF BACHELOR OF SCIENCE OF ENGINEERING AT UNIVERSITY OF MORATUWA, SRI LANKA

March 28, 2017

Declaration

This thesis is a report of the project work and research carried out in the Department of Computer Science and Engineering, University of Moratuwa, from 4th April 2016 to 30th of March 2017. Except where references are made to other work, the context of the thesis is original and includes the work done in collaboration as a team. This thesis has not been submitted to any other university.

Signatures of the candidates:

Signature	Name and Index Number	Date
1.	G.H.D.M.M. Gammanpila (120165D)	....
2.	A.M.M. Gangananda (120168N)	....
3.	B.S. Kalansuriya (120282H)	....
4.	W.D.T. Piyadasun (120478N)	....

Signature of the project supervisor:

Signature	Name	Date
....	Dr. H.M.N. Dilum Bandara	....

Abstract

Crimes have a negative effect on any society both socially and economically. Law enforcement bodies face numerous challenges while trying to prevent crimes. We propose a Crime Data Analytic Platform (CDAP) to assist law enforcement bodies to perform descriptive, predictive, and prescriptive analysis on crime data. CDAP has a modular architecture where each component is built separate from each other. CDAP also supports plugins enabling future feature expansions. The platform can ingest any crime dataset which has the required attributes to map dataset to attributes required by the platform. It can then analyze them, train models, and then visualize data. CDAP also combines census data with crime data to achieve more comprehensive crime analysis and their impact on society. Moreover, with the combination of census data and crime data, CDAP provides process reengineering steps to optimize resource allocations of police forces. We demonstrate the utility of the platform by visualizing spatial and temporal relationships in a set of real-world crime datasets. Predictive capabilities of the platform are demonstrated by predicting crime categories, for which a machine learning approach is used. To construct a model Nave Bayesian, Random Forest Classifier, and Multi-layer Perceptron Network classification algorithms are provided. Identification of optimized police district boundaries and allocating patrol beats are used to demonstrate the prescriptive analytics capabilities of the tool. Heuristic-based clustering approach was taken to define police district boundaries in a way that the identified districts have equitable population distribution with compact shape. The resulting districts are then evaluated on inequality of population and the compactness using Gini Coefficient and Isoperimetric Quotient. Another heuristic-based approach was taken to define new police patrol beats to be optimized on equitable workload distribution, compactness, and minimizing response time for new police patrol beats.

Acknowledgements

First and foremost we would like to express our sincere gratitude to our project supervisor, Dr. H.M.N. Dilum Bandara for always extending fullest cooperation and dedicated guidance throughout the project

We would also like to thank our external supervisor Dr. Suresh Marru for valuable insight and direction shared regarding our project. In addition, we would like to thank Dr. Surangika Ranathunga and Ms. Sandareka Fernando for being panel members for our evaluations and providing us with pointers for improvements.

We would also like to thank Dr. Charith Chitraranjan for coordinating final year projects successfully.

And last but not least we would like to extend our sincere gratitude towards Department of Computer Science and Engineering, University of Moratuwa for providing the support for us to successfully finish the project.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Background

Crimes are a social nuisance and it has a direct effect on a society. Governments spend lots of money through law enforcement agencies to try and stop crimes from taking place. Today, many law enforcement bodies have large volumes of data related to crimes, which need to be processed to turn into useful information [3]. Crime data are complex because they have many dimensions and in different formats, e.g., most of them contain string records and narrative records. Due to this diversity, it is difficult to mine them using off the shelf, statistical and machine learning data analytics tools.It is the primary reason for lack of general platform for crime data mining. While there are some propitiatory platforms to predict and analyze crime data, they are focused only on certain areas of crimes, not extensible, and do not provide an API to integrate with other tools [4]. Moreover, the same tool cannot be used for the analysis and well as planning such as patrol beads and district boundaries.

1.2 Motivation

High or increased crime-levels make communities decline, as crimes reduce house prices, neighborhood satisfaction, and the desire to move in a negative manner [5]. To reduce and prevent crimes it is important to identify the reasons behind crimes, predict crimes, and prescribe solutions. Due to large volumes of data and the number of algorithms needed to be applied on crime data, it is unrealistic to do a manual analysis. Therefore, it is necessary to have a platform which is capable of applying any algorithm required to do a descriptive, predictive, and prescriptive analysis on large volume of crime data. Through those three methodologies law-enforcement authorities will be able to take suitable actions to prevent the crimes. Moreover, by predicting the highly likely targets to be attacked, during a specific period of time and specific geographical location, police will be able to identify better ways to deploy the limited resources and also to find and fix the problems leading to crimes.

Several applications are already developed for crime analysis [6] [7]. Most of these tools are developed to help the police to identify different crime patterns and even to predict criminal activities. They are complex software which needs a lot of training before use. Designing a tool which is easy to use with minimal training would help law-enforcing bodies all around the world to reduce crimes.

1.3 Problem Statement

The research problem that this project try to address can be stated as follows:

How to develop a software platform to conduct descriptive, predictive, and
prescriptive analysis of diverse crime data?

Descriptive analyzing focuses on identifying spatial temporal relationships with crime data.Predictive analytics methods are mainly used for predicting category of a crime which can be occurred somewhere at a given time.In order to achieve it system integrate Census data with the crime data and feed it to machine learning algorithms.In prescriptive analyzer it suggests process re-engineering steps to allocate police resources optimally with the intention of reduce crimes and impact to the general public.

1.4 Research Objectives

Research objectives of this project are as follows:

- Develop a platform that can be used to analyze crime data using descriptive and predictive data analytics techniques.
- Using the proposed platform analyze the spatial and temporal (time of day, day of week, and seasons) relationships in crime data.
- Suggest suitable process reengineering steps and resource allocations based on the spatial and temporal relationships. For example,
  - Identify new police district boundaries using Heuristic-Based Polygonal Clustering methodology.
  - Identifying intelligent patrol routes that can combine crime data and spatial dimensions using Voronoi Tessellations.
- Analyze relationship between crime data and census data.

1.5 Outline

The rest of the report is organized as follows. Chapter 2 presents related work. Design of the proposed platform is presented in Chapter 3. Chapter 4 describes implementation details of the platform as well as implementation of predictive and prescriptive features. Summary and future work is presented in Chapter 5.

2. Literature review

Section 2.1 defines crimes and related theories. Existing platforms and importance of a platform which is able to carry out descriptive, predictive, and prescriptive analytics of crime data are discussed in Section 2.2. Section 2.3 discusses about the infrastructure, tools, algorithms, visualization, and different types of architectures related to the solution.

2.1 Crimes

2.1.1 Crimes and Effect on the Society

A crime can be defined as any action or omission that violates a law, which results in a punishment. Usually what constitutes as a crime depends on the government bodies and laws that are in existence in those places. To understand the nature of crimes, one has to understand not only its spatio-temporal dimensions, but also the nature of the crime, the victim-offender relationship, role of guardians, and the history of similar incidents [8].Regardless of the reasons why crimes take place, they put a strain on the communities, towns, and cities. Usual monetary costs associated with them include cost of policing crime and prosecuting those who commit crimes. Non-monetary costs consist of social costs, where they affect the quality of life, mental health, and physical security of people living in those areas. Crimes are a social nuisance and being able to solve them faster is very important and will pay for itself [4].

2.1.2 Criminology Theories

According to John and David [9], theories of crimes can be divided into two categories namely, those that seek to explain the development of criminal offenders and those that seek to explain the development of criminal events. Criminology has been mainly developed through theories and research on offenders. Only recently it has begun to explain the crimes rather than criminality of people involved in it. Criminology consists of many theories that explain how and why some offenders act in the way they do. Following are some of theories that explain how places are associated with crimes [9].

1. Rational Choice Rational Choice suggests that offenders will select targets and define means to achieve their goals in a manner that can be explained. Further it can be explained as that human actions are based on rational decisions, that is they are informed by probable consequences of that action [9].

2. Routine Activity Theory This theory explains the occurrence of crimes as the result of several circumstances. Namely, a motivated offender, a desirable target, target and offender must be at the same place at the same time, and lastly absent of other types of controllers intimate handlers, guardians, and place managers [9].

3. Crime Pattern Theory This theory combines the above two theories and goes on to say that how targets come to the attention of offenders is influenced by distribution of crime events over time, space, and among targets. An offender will come to know of criminal opportunities while engaging in their day-to-day legitimate work. So a given offender will only know about a subset of available targets. The concept of place is essential to crime pattern theory [9].

Having an understanding of criminology theories is essential to try and create crime analysis tools or platform using modern technologies.

2.1.3 Crime Analysis

Crime analysis is a difficult task, as it requires both collection and analysis of large volumes of data. For example, Brown [6] states that Richmond city in USA has approximately 100,000 criminal records per year. Given the data volume and need to apply different algorithmic techniques forbids manual analysis. Whereas an automated analysis of such a rich data set could identify complex crime patterns and assist in solving crimes faster.

Data mining techniques can be used in law and enforcement for crime data analysis, criminal career analysis, bank fraud analysis, and analysis of other critical problems [8]. Some of the traditional data mining techniques are association analysis, classification and prediction, cluster analysis, and outlier analysis, which identify patterns in structured data [10]. By using criminology theories along with modern technology would help to identify crime patterns quickly and efficiently. To simplify the workload a crime data analytic platform could be used which would help in simplifying the process while driving more accurate and insightful conclusions and predictions.

2.2 Existing Platform

Several applications have been already developed for the purpose of crime analysis. Most of these tools are developed to help the police forces to identify different crime patterns and even to predict criminal activities. Recent applications were developed by aiming at adopting data mining techniques. Next, we discuss some of key solutions [5].

1. COPLINK

   Chen et al. [7] describes COPLINK as an integrated information and knowledge management environment trying to manage massive amount of information on criminals. It has been developed at the University of Arizonas Artificial Intelligence Lab in collaboration with the Tucson and Phoenix Police departments. The main aim of this project is to develop information and knowledge management systems, technologies, and methodologies appropriate for capturing, accessing, analyzing, visualizing, and sharing law enforcement-related information in social and organizational contexts. COPLINK consists of two main parts namely, COPLINK connect and COPLINK detect. COPLINK connect is responsible for sharing of data from different police departments, while COPLINK detect is used to uncover different crime associations that exist in police databases. Detect is mostly concerned with creating associations and linkages among various aspects of a crime. It uses statistical techniques such as co-occurrence analysis and clustering functions to weight relationships between all possible pairs of concepts.However, following drawbacks can be identified in this system [7].

   (a) COPLINK is complex and requires user training.

   (b) Although the system is able to identify linkages among specific concepts residing in existing database, it does not support data mining.

   (c) Current version of COPLINK does not support temporal reasoning or visualization.

2. ReCAP

   Regional Crime Analysis Program (ReCAP) [6] is a system designed to aid local police forces through crime analysis and prevention. It works with Pistol 2000 records management system. The system is quite old and is based on Windows 95 and NT, and works only with a Local Area Network (LAN). This system has three main parts, namely a database, geographic information system, and data mining tools. It provides following key functions.

   (a) Hot Sheet gives a summary of most important crime activity of the region.

   (b) Summary Reports These reports tally specific crime occurrences over a user-defined time and area.

(c) Map-oriented Searches Provide a GIS display of the area along with plotted criminal activity. The crimes can be displayed on the map according to the type of crime, time/date of crime, location, suspect description, weapons involved, etc.

(d) Time Charting Patterns developing over time of the day, day of the week, etc., are plotted. Indeterminate crimes are plotted statistically using kernel density estimation.

(e) Cluster Analysis Uses algorithms such as k-means and nearest neighbor to perform clustering, to identify statistically significant groupings of crimes in an area.

(f) Detailed Inter-Modular Searching Helps in detecting links between vehicles, suspects, warrants, etc.

3. Crime Prediction Model

   Another solution is Ozguls Crime Prediction Model (CPM) [5], which predicts offenders of terrorist events based on location, date, and modus operandi attributes. It uses both solved and unsolved crime information, learning from attributes of each crime. These crimes are clustered according to the attributes. As an example, similar located crime clusters. These clusters contain crimes according to specific attributes. Similarity scores are measured for each crime and total similarity between two crimes is measured using Euclidean distance. Using these values similarly behaving solved and unsolved crimes are put into more accurate clusters. CPM looks for perpetrators of crimes, with the assumption that majority of crimes in a single cluster were committed by the same single offender group. This platform focuses mainly on terrorist activities and groups.

4. Rationalize police patrol Beats using Voronoi tessellation

   Several related work focus on optimizing available resources to reduce crimes. Suresh et. al. [8] demonstrates how to use Voronoi Tessellations to divide a given police jurisdiction into a set of patrol beats for equitable workload. As a proof of concept, they have used a sample data set from Indianapolis police department. Different weights were assigned to crimes according to their severity. Then Police beats were distributed according to crime types, crime data, crime groups, and geography. As future work, authors have proposed to extend the Voronoi tessellations to meet different requirements of the police. One such consideration is to rationalize the boundaries of police beats by taking road network and physical landscapes like rivers into account. This can be also extended by considering data like offender residencies for better surveillance. However, authors have not implemented a framework or any kind of implementation on the theories discussed other than a proof of concept.

5. Optimal Selection of Police Patrol Beats

Mitchell [11] proposed a model for select patrol beats based on heuristic approach.It takes some amount of assumptions take into account while proposing the model. Among those assumptions the incident distribution, over both space and time, is assumed, and that a distance measure or metric between the centers of each subunit assumed and also the nearest available unit responds to a call.

A heuristic based approach was used in model building where the following heuristic was used.

$$minimizeA\sum minimumW(i,j) \tag{2.1}$$

W(i,j) is the matrix of appropriately weighted distances. The assumptions of the model require that W(i, j) be a distance matrix with the (i, j)-th element representing the weighted travel or other distance from the i-th location to the j-th location. The objective is then to choose a subset of k rows of W in such fashion as to minimize the sum of the column minimums, where each column minimum is chosen only from among the designated subset of k rows.The heuristic algorithm Mitchel proposed has two phases. In the first phase, k locations are selected in some fashion. In next improvement phase the algorithm seeks to improve on locations selected in the first phase, by sequential substitution of the locations selected. The process is repeated for each of the locations not in the allocation until no improvement is made after a complete cycle [11]

There are several other platforms and models described in several papers about crime data analysis. Revathy and Satheesh [5] mentioned several other solutions like Self Organizing Map (SOM) which links sexual offenders of sexual attacks. Each of the above platforms and solutions assist law-enforcement bodies to analyse and identify different crime patterns. One of the main reasons for developing of different platforms to analyse crime data is, the huge volume of data that is needed to be analysed. This task has become impossible to do manually. A lot of research is done on ways to identify crime patterns using different algorithmic methods like clustering used in Crime Prediction Model. Taking these available resources into usage is very important. Providing a single platform which is capable of using different techniques used in different platforms is important to analyse and identify crime patterns. By using predictive models, authorities are able to identify which kind of crimes could occur most in a given time period around which areas. Identifying these details is very important for different law enforcement authorities to make decisions on how to minimize crime. A platform being able to analyse different crime patterns descriptively would help to identify patterns in crime and there are some platforms which already provide this facility [6,7]. Being able to predict the type of crimes which could occur in a given area at a given time is also very useful and prediction is used by crime prediction model to identify terrorists involved in a terrorist activity. A platform being able to provide prescriptive analysis on ways to minimize crimes could stop crimes from happening. This could also help law enforcement authorities to make use of their limited resources in the most effective way.

After going through these solutions, it is clear that these platforms are specific for a given task. It is very useful to have all the above mentioned analytical techniques in one platform. That is a platform which can be extended to provide descriptive, predictive and prescriptive analysis of crime data.

2.3 Tools, Algorithms, and Infrastructure

2.3.1 Tools

1. Apache Storm

   Apache Storm is a free and open source distributed real time computation system. Storm processes large volumes of high-velocity data in real-time. It is extremely fast and can process over a million records per second per node on a cluster of modest size. Storm on Hadoop YARN (Yet Another Resource Negotiator) is powerful for machine learning purposes and mainly for real-time analysis. Some of the use cases of Storm are real time analytics, online machine learning, continuous computation, distributed RPC, ETL, and more. Apache Storm operates on continuous stream of data which is not going to happen in our use case for crime analysis. Apache Spark performs Data-parallel computations while Storm performs Task-Parallel computations [12].

2. Apache Spark

   Apache Spark is a general purpose cluster computing engine which is considered very fast and reliable [13]. It is an open-source platform for large-scale data processing that is suitable for iterative machine learning tasks [14]. Apache Spark provides Application Programming Interfaces (APIs) in programming languages such as Java, Python, and Scala. Spark provides generality by powering a stack of libraries which includes SQL and Data Frames, MLlib for machine learning, GraphX, and Spark Streaming.These libraries are built upon the Apache Spark core and they can seamlessly combine in the same application. Spark runs on Hadoop, Mesos, standalone, or even in the cloud. Spark supports in-memory computing which has enabled it to query data much faster compared to disk based engines such as Hadoop. Apache Spark website gives statistics that prove this fact. It shows that Spark runs programs up to 100x faster than Hadoop MapReduce in memory and 10x faster on disk. Spark can analyze large volumes of crime data, which is required in our framework. Reasons to choose Spark [15]:

(a) Spark uses Resilient Distributed Datasets (RDDs) which allows storing data on memory and persist it as per the requirements. It allows a massive increase in batch processing job performance.

(b) Spark allows to cache data in memory, which is useful in case of iterative algorithms that are used in machine learning. It is also important because it is very useful for interactive data mining which is required the proposed platform.

(c) It provides a feature to join datasets across multiple disparate data sources.

(d) When comparing time taken for k-means algorithm on a datasets of different sizes K-means using Spark always gave much better performance than K-means using MapReduce.

For Machine Learning purposes Apache Spark provides MLlib, a distributed machine learning library. It consists of many fast and scalable implementations of standard learning algorithms for common learning settings including classification, regression, collaborative filtering, clustering, and dimensionality reduction. It also provides some underlying statistics, linear algebra, and optimization primitives. MLlib includes Java, Scala, and Python APIs. Because of Spark Integration, this library can use core functionalities for purposes like data cleaning and featurization. Spark SQL provides data integration functionality. When compared performance wise with apache Spark, Apache Mahout v0.9 on Hadoop MapReduce was much slower mainly due to MapReduces scheduling overhead and lack of support for iterative computation [14]. Apart from all this, Apache Spark has extensive documentation and a large and active community. Considering all the above factors we chose Apache Spark as a tool for our project.

2.3.2 Algorithms

Following are the algorithms which can be used in the proposed solution.

1. Clustering Algorithms

   Several papers have mentioned about many available algorithms that can be used for the purpose of analyzing crime data. Shyam Varan Nath says in his paper that 10% of criminals commit 50% of the crimes. Data mining has a higher influence in fields such as Law and Enforcement for crime problems, crime data analysis, criminal career analysis, bank frauds and other critical problems [1]. Following are some common clustering algorithms that can be used for crime data analysis. Shyams paper goes on to suggest that clustering technique is a better approach than any other supervised techniques such as classification since crimes vary in nature widely and crime database often contains several unsolved crimes. Also nature of crimes changes over time, so in order to identify newer and unknown patterns in future, clustering techniques work better [5].

   (a) K-means Clustering Algorithm

   This algorithm is mainly used to partition the clusters based on their mean. As a first step number of objects are grouped and specified as k clusters. K numbers of objects are initially selected as the cluster centers. Then again these objects are assigned based on cluster center. Then cluster means are updated again. This algorithm is used as a base for most of the other clustering algorithms [1].

   (b) AK-mod Algorithm

   This is a clustering algorithm consisting of two phases. In the first phase attributes are weighted. Weights of the attributes are calculated using the Information Gain Ratio(IGR) for each attribute. The attribute with the greatest value is taken as the decisive attribute. In the second phase (clustering) first the number of clusters k and initial mode of each cluster are found. Then distance for every mode and its closest mode are calculated. After that each cluster mode is updated. This process keeps going till all modes are not updated again [1].

(c) Expectation-Maximization Algorithm

This is an extension of k-means clustering algorithm. It is used to calculate parameter estimates for each cluster. Weights of attributes are measured in probability distribution and each object is to be clustered based on the weights. To measure parameter estimates two steps are followed.

i. Expectation Step: In this step, for each object of clusters the probability of cluster membership of object *x(i)* is calculated.

ii. Maximization Step: Re-estimate/refine model parameters using estimation from step one [1].

2. Classification Algorithms

Classification is considered as a supervised prediction technique. It has been used in many domains like weather forecasting, health care, medical, financial, etc. Two different classification algorithms are considered. They are namely Decision Tree and Naive Bayesian. Naive Bayesian is considered as an effective algorithm to solve classification tasks. Decision tree is a commonly used predictive model and it also follows supervised learning approach. As the name suggests it forms a tree like structure and each node represents a test on attribute value. Leaves represent classes that predict model for classification. Branches represent conjunctions of features. This algorithm applies a top down approach. Gain in entropy is used to guide the algorithm for creation of nodes. There are some pros and cons of both algorithms. Naive Bayesian requires a shorter training time and it has a fast evaluation. It is more suitable for real world problems. However, for complex classification problems decision tree is more suited. It produces reasonable and interpretable classification trees. This paper suggests that Decision Tree has a higher accuracy and precision over Naive Bayesian [16].

(a) Decision Tree

Decision tree learning is a method commonly used in data mining and machine learning tasks, classification and regression. Decision trees break down a dataset into smaller subsets while at the same time an association tree is incrementally developed. Final tree has decision nodes and leaf nodes. Decision nodes have two or more branches while leaf node represents classification or decision. Decision trees can handle both categorical and numerical data. The core algorithm for building decision tree is called ID3 which employs a top-down, greedy search through the space of possible branches with no backtracking. Each partition is chosen greedily by selecting the best split from a set of possible splits, in order to maximize the information gain at a tree node. Information gain is the difference between the parent node impurity and the weighted sum of the two child node impurities. Impurity can be measured using Entropy. $C_1^p - f_i \log f_i$ equation calculates the entropy. Here, C is the number of unique labels, f_i is the frequency of label i at a node [17].

Advantages of using decision tree algorithm are [18]

i. It is simple to understand and interpret.

ii. This technique requires very little data preparation.

iii. Decision trees can handle both numerical and categorical data.

iv. Decision trees perform well with large datasets. It can be used to analyze large datasets within reasonable time using standard computing resources.

(b) Random Forest Classification

Random forests are one of the most successful machine learning models for classification and regression. Random forests algorithm does not over fit. It is achieved by combining many decisions trees. One can run as many trees as needed. Random forests algorithm is considered fast. Basic algorithm of Random forests trains a set of decision trees separately, so the training can be done in parallel. The algorithm injects randomness into the training process to make each decision tree different from each other. By combining predictions from each tree reduces the variance of predictions, improving the performance on test data. Following are some of the features of Random forests classification [19].

i. Random forests handle categorical features and can even extend to multi class classification.

ii. It is able to capture non-linearity and feature interactions.

iii. This algorithm runs efficiently on large data bases.

iv. It can handle thousands of variables without variable deletion.

v. Random Forest algorithm is effective for estimating missing data and maintains accuracy even when large parts of data are missing [19, 20].

(c) Multi-Layer Perceptrons (MLPs)

Multi-Layer Perceptrons (MLPs) are classified as a type of Artificial Neural Network. The computation is performed using set of many simple units with weighted connections between them. Multi-layer perceptron consists of some layers. They are namely,

i. Input layer: This is the bottom most layer that takes input from a dataset. It is the exposed part of the network.

ii. Hidden layers: One or more layers which are not directly exposed to the input.

iii. Output layer: takes the output from the final hidden layer. It outputs a value or vector of values that correspond to the format required for the problem [21].

3. Frequent Pattern- Growth (FP-Growth) Algorithm

In this algorithm the first step is to calculate item frequencies and identify frequent items. There is a tree- structure called Frequent-Pattern tree used for the next step. Frequent pattern tree is a tree structure defined as below. It consists of one root labeled null, set of item prefix subtrees as the children of the root and a frequent-item header table. Each node in the item prefix subtree consists of three fields: item-name, count, and node-link. Each item in the frequent-item header table consists of two fields, namely item-name and head of node-link, which points to the first node in the FP-tree carrying the itemname. FP-growth algorithm uses this structure to encode transactions without generating candidate sets explicitly, which is considered as an expensive process. After this process, frequent itemsets can be extracted from the FP-tree.

There are advantages of using FP-Growth algorithm.

(a) It has only two passes over the dataset

(b) It compresses the dataset.

(c) FP-Growth algorithm is much faster than Apriori.

2.3.3 Visualization

It is important to visualize the crime data in order to get a clear idea about its distribution and also to display the analyzed results in a user friendly manner. When achieving this task it is important to display time dimension within GIS visualization system. But more focus has been given to display the spatial distribution of high crime areas or hot spots of crime. An example of this is pin maps, which have been used since the beginning of modern police systems. The utility of crime maps has increased significantly since the growth of Geographic Information Systems (GIS) and geo-archives that link criminal data with socio-economic and environmental data such as location of high schools, liquor shops or metro routes that potentially impact crime. A new range of statistical and simulation modeling techniques are proving very useful in spatial analyses of crime sites and patterns through computer mapping. Examples of statistical and simulation modeling techniques include point pattern analysis, nearest neighborhood analysis, location quotient analysis , travel pattern analysis, neural networks and numerous statistical techniques including cluster analysis, spatial regression and auto correlation For example, using these tools, investigators are gaining insight into the location behavior of offenders. Moreover, new developments in criminological theories like environmental criminology require more reliance on visual graphical methods such as topology, and fractal geometry so that patterns, designs and edges stand out. However, almost all crime mapping efforts and techniques has focused on spatial domains typically by aggregating the crime data over time. So it is important to identify techniques to display crime data in a way that spatio-temporal attributes of crimes are presented [1].

1. Pin Maps

   A pin map can give the viewer a very good feel for the density and distribution of crime. The viewer gets a very good overall picture, without losing sight of the individual crimes. These pin maps can be rotated, zoomed in, translated and tilted to examine the data more easily from any angle.Figure2.1 shows how pin maps appears when zoomed in. Furthermore, different variants of pin maps can be created easily. Perhaps the greatest limitation of a pin map is that it is a purely spatial display. In order to produce a pin map, the user has to pre specified a time zone for which the pin map is created.

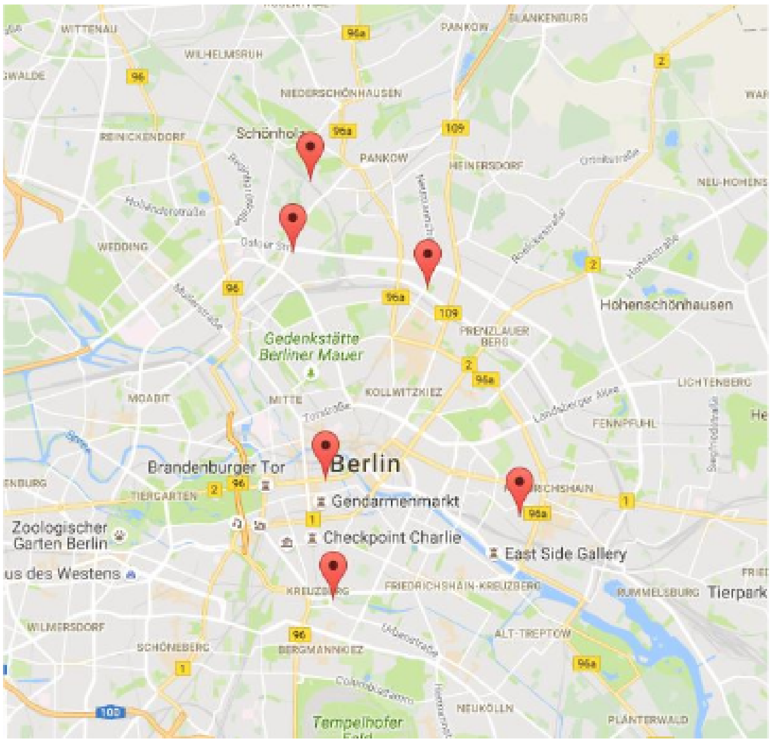


Figure 2.1: Pin map

2. Pseudo-Colored Time-Window Display This technique can be used to represent

data in two-dimensional system. In this technique user pre specifies a time zone as in a pin map. Total crime within a grid cell for the specified time zone is then aggregated and shown as a color on a two-dimensional display by mapping data to a color using some pseudo-coloring scheme as shown in Figure 2.2. In this cumulative variation, the pseudo-coloring visualization is simply a version of a pin map. In the time-window scheme, the user can specify a time interval and the two-dimensional display generates a rolling animation of pseudo-colored crimes within the specified interval as time advances. This technique is useful in depicting the shifting of hot spots of crime as time advances.

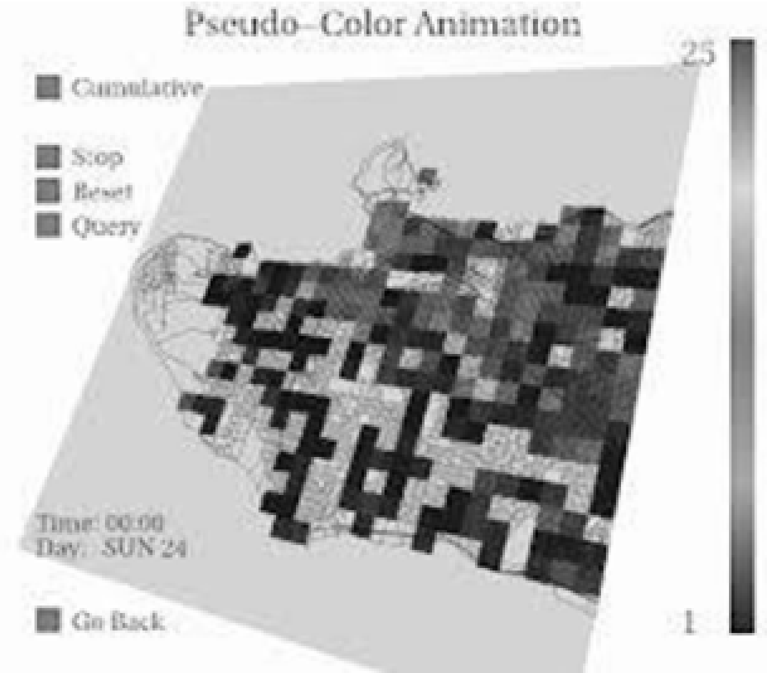


Figure 2.2: Pseudo colored time-window display [1]

3. Side-by-Side Height Bars

This three dimensional display technique is helpful in comparing the spatial data pertaining to two different time zones. In this technique one type of bars indicates the crimes committed during one time frame of day while other type of bars indicates the crimes committed during other time frame of the day. This view is useful in allocating police personnel to areas during different patrol shifts. One advantage of this method is the familiarity of the user with height bars in two-dimensional displays. Depending upon the cell size, this technique can be extended to compare more than two time zones.Figure 2.3 shows a sample visualization which has used side by side height bars.

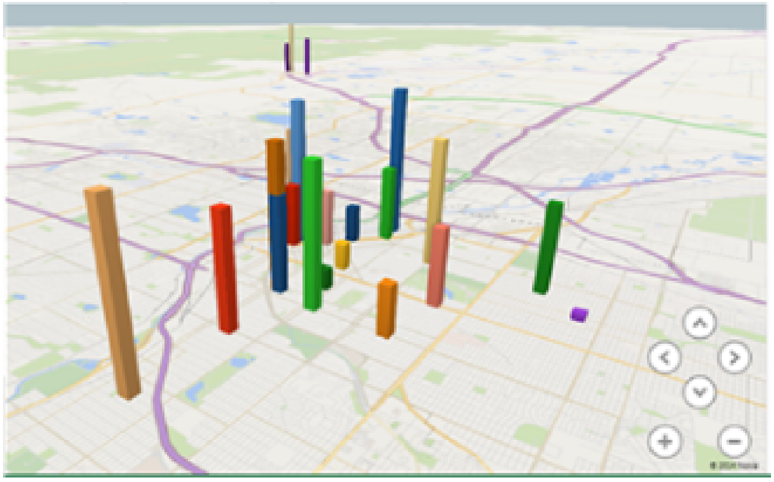


Figure 2.3: Side-by-side height bars [1]

4. Stacked Time-Aggregated Cumulative Bars in three dimensions

   This technique can be thought of as a three dimensional version of a pin map as shown in Figure 2.4. As in a pin map, the user pre specifies a time zone. All the crimes of a specified type for the time zone are then aggregated and stacked over each other in three dimensions. The main advantage over the pin map is the reduction of clutter and addition of at least one more piece of information in a very convenient manner in this case, the crime type. This technique is helpful in highlighting the hot spots clearly by depicting them as high-rise spot. Here the user can choose to display any combination of crime types.

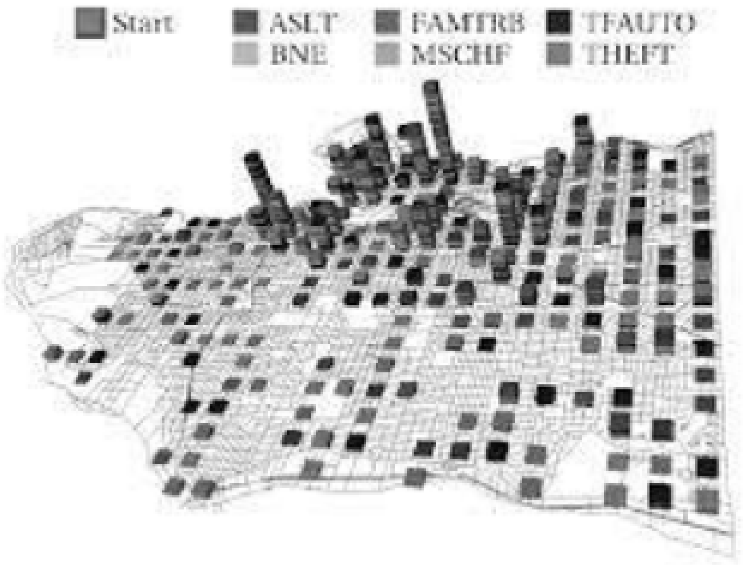


Figure 2.4: Stacked Time Aggregated Cummilative Graph [1]

5. Space-time Cube

   Idea of this is to create a three-dimensional map of crime events in a space-time cube to visually and simultaneously comprehend the temporal duration as well as the spatial extent of different crime clusters rather than focusing on an individual cluster as Figure 2.5. A space-time cube is a 3D space consisting of two horizontal dimensions of space (geographic plane) and one vertical dimension of time [22].

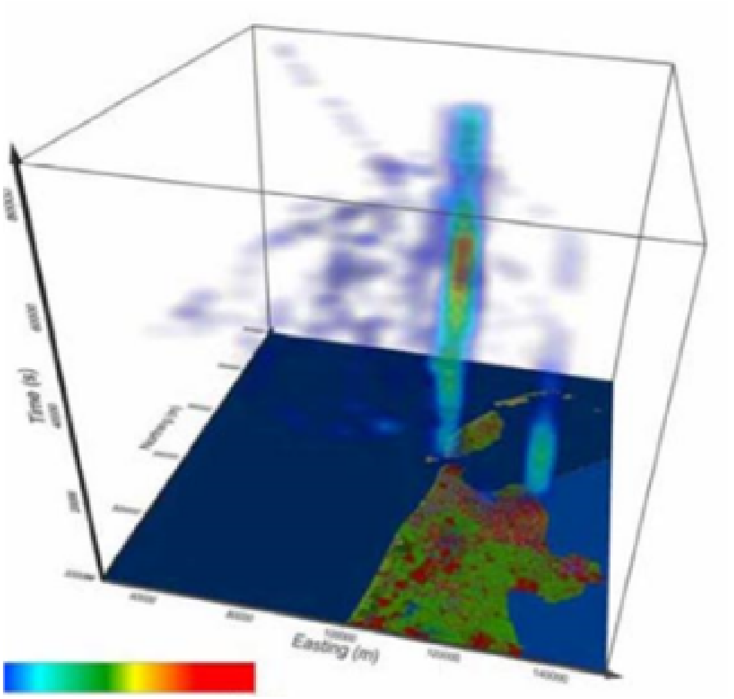


Figure 2.5: Space Time Cube Visualization [1]

6. Hot Spots

   Crime hot spots are prime exemplars of the potential value of place in the analysis of crime. Using this technique we can identify highly concentrated crime areas. Importance of this is we can use that to have an idea about expected dispatched police calls for service. While often motivated by pragmatic concerns about what interventions are likely to be effective in reducing crime, results like these also serve to sharply focus crime theory on developing satisfactory accounts of these apparently strong relationships between crime and place. Crime studies that examine the spatial distribution of crime clearly demonstrate that certain land uses and population characteristics are associated with crime hot spots. Hot spots are systematic (regular and predictable) and not just random occurrences. Figure 2.6 displays a heatmap for crime locations within San Francisco.

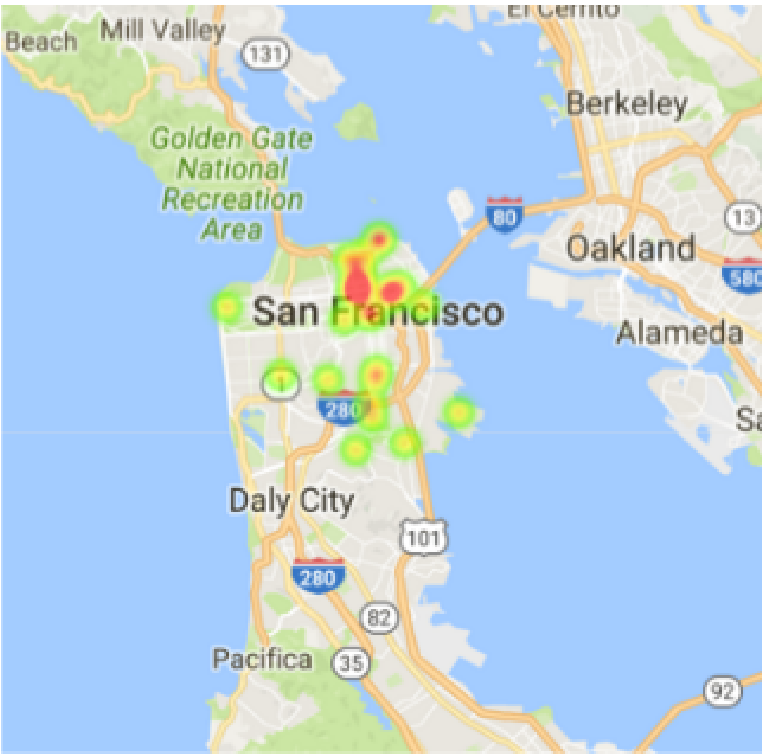


Figure 2.6: Side-by-side height bars

2.3.4 Architectures

1. Lambda Architecture

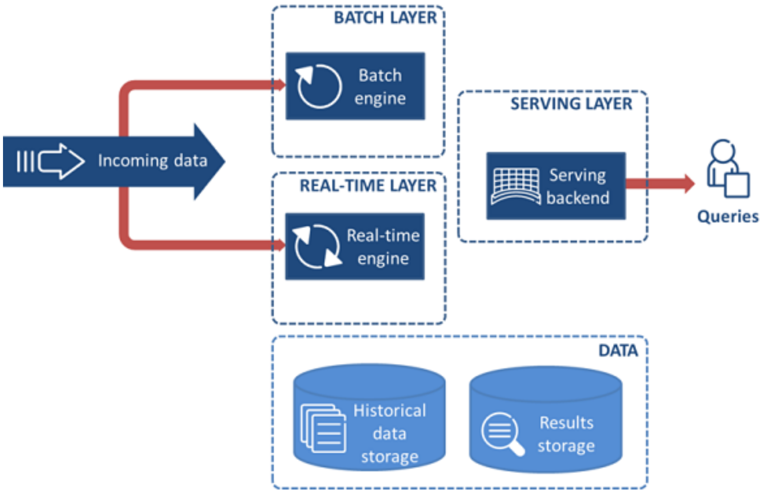


Figure 2.7: Lambda Architecture [2]

Lambda architecture which is shown in Figure 2.7, is a generic, scalable and fault-tolerant data processing architecture. It is a combination of speed layer, batch layer and serving layer. All the data entered to the system is dispatched to both batch layer and speed layer for processing [2].

Data dispatched to the batch layer is deposited in the master data set and processed periodically with a delay and considered as complete because it retrieved all the historical data. Using that historical data batch processing component builds machine learning models. Data dispatched to the speed layer is processed that data real time and generate real time views. But these real time views not considered as complete. Serving layer depends on the use case of the application it takes action according to the things happening in the batch layer and service layer. Any incoming query can be answered by merging results from batch views and real time views.

Applying a complex architecture like lambda architecture to the crime data analytic platform which does not require real time data processing is controversial. Crime records are kept as dataset which contain information about several domains collected over years.

So Crime data does not need stream to the system. They can be fed to the system as historical data sets and it does not require a real time data processing.

2. Kappa Architecture

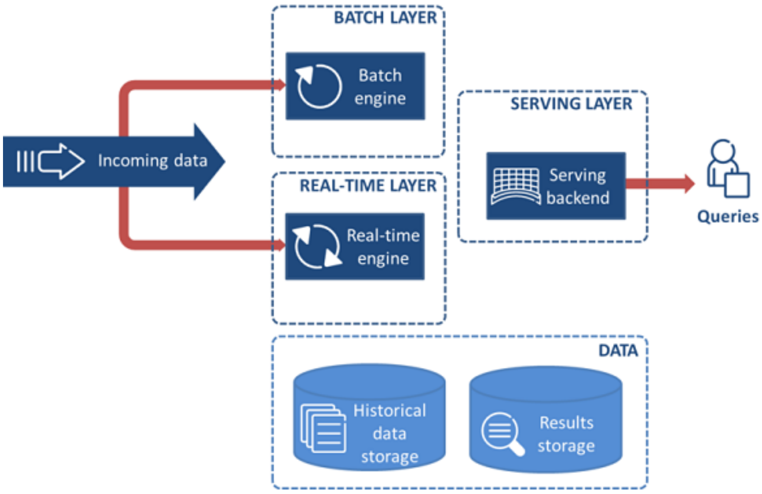


Figure 2.8: Kappa Architecture [2]

Kappa Architecture is a simplification of Lambda Architecture. A Kappa Architecture system is like a Lambda Architecture system with the batch processing system removed. To replace batch processing, data is simply fed through the streaming system quickly.

Kappa Architecture is centered on a unified Log, all the streaming data will be moved as unified log meaning a single deployment where every event collected is immutable, ordered, and the current state of the event will be changed only by appending a new event.As shown in Figure 2.8, all the processing of the event will be performed in the input streams and persisted as real-time views. In order to support human fault tolerant, the events are also persisted in storage like HDFS if the data is aged out of the unified logs [2].

In real time in order to reprocess an event start a second instance of the job which starts processing from the beginning of the event and direct the output to a different table, when the reprocessing job is caught up point the application to the new table and drop the old table. This continues in the streaming layer if something needs to be reprocessed without having the Batch Layer.

The advantage of this architecture is, the developer needs to reprocess only when the code is changed. If the changed code does not work fine he can roll back to the old output table. Also it is possible to mirror the kafka topics to HDFS storage for long term use.

3. Proposed Design

3.1 Solution

The platform which is going to be developed directly targets the crime domain. It has the ability to analyze crime data in three different ways, namely Descriptive, prescriptive, and predictive analytics. The most important part of this platform is that, it is designed to be scalable to support different types of crime data analysis. Different user requirements can be achieved through developing simple plug-ins to the system and scale the platform.

Descriptive Analyzer use both quantitative and qualitative data along with analytical techniques. Qualitative data and analytical techniques refer to non-numerical data, as well as the examination and interpretation of observations for the purpose of discovering underlying meanings and patterns of relationships. Descriptive analyzer basically provides relationships between crimes and identifies the pattern of crimes and temporal and spatial relationships between crimes. It also provides statistical summary of a given data set. Prescriptive analyser tries to identify the reason behind the crimes and gives the suggestions to avoid or reduce the crimes. It has the ability to identify significant factors related to the crimes committed. Through plugins users can manipulate and extend the platform for specific needs.

In this platform, predictive analytics methods are mainly used for predicting category of a crime which can be occurred somewhere at a given time. Predict crime category system integrates population and race data accordingly to the given crime data set. Using machine learning techniques it can predict the category of crime that can occur. Any user or a law enforcement body, who has crime dataset can use this feature to understand the severity of crime that would take place, and as a result could take necessary steps to allocate resources effectively. Following sections describe about the architecture and different modules implemented in the solution.

3.1.1 Features provided by the platform

The following features are provided by the CDAP.

1. Redraw efficient police jurisdiction boundaries.
2. Query data in the crime dataset.
3. Draw efficient police patrol beats based on the crime distribution.
4. Predict crime categories for a given crime scenario.
5. Upload a crime dataset and population dataset.
6. Preprocess the uploaded crime dataset.
7. Discover spatial and temporal patterns of crime data using visualization features.

3.2 Architecture

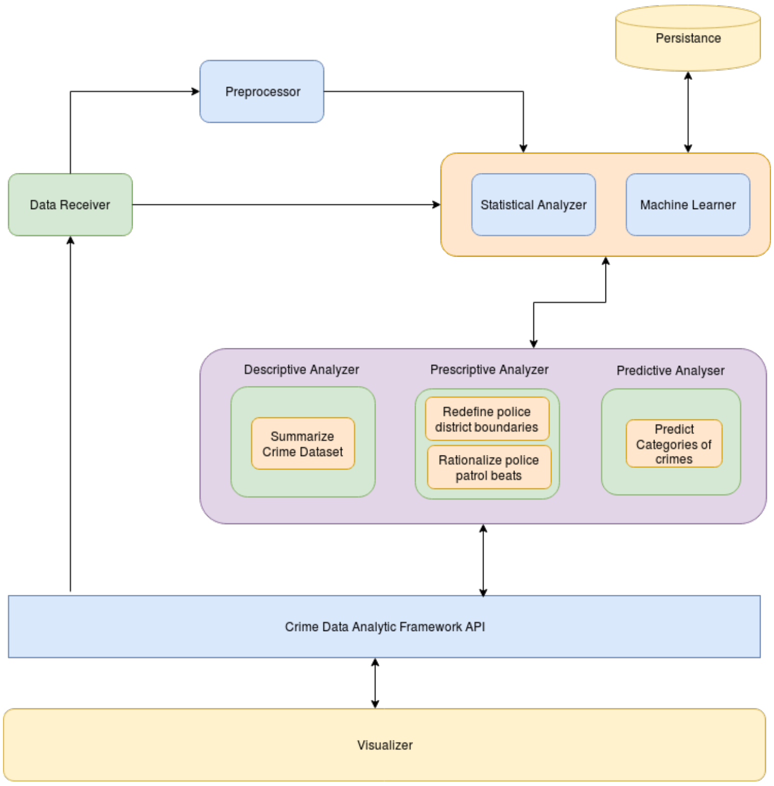


Figure 3.1: High-level architecture of the proposed platform

Figure 3.1 depicts the high-level architecture of the proposed Inquisitors Crime Data Analytic Platform (CDAP). Data Receiver is used to provide data to the platform and the data persistence unit is used to store data used by the system and trained models. The preprocessor is used to preprocess raw data received by the Data Receiver which in turn fed into the next layer. It consists of two core modules, Statistical Analyzer and Machine Learner. Using those two modules, the Descriptive analyzer, Prescriptive analyzer, and Predictive analyzer have been implemented. Also there are few other components which provide a vital contribution to the CDPA. The functionalities which are provided by prescriptive, descriptive and predictive analyzers are exposed to the user through an API. The results which are provided by CDAP according to the users requests can be visualized using the Visualizer component.

3.2.1 Preprocessor

The module Preprocessor which is used by all other modules, consists of the features and functionalities which are needed to preprocess the data before feeding them into other modules such as Machine Learner, Statistical Analyzer.

Within this module, functionalities needed to preprocess the input data have been implemented. After receiving the CSV data file from the user specified location, the data extracted from that file will be processed in three stages.

1. Data Cleaning: Data is cleansed through processes such as filling in missing values, smoothing the noisy data, or resolving the inconsistencies in the data.
2. Data Integration: Data with different representations are put together and conflicts within the data are resolved.
3. Data Transformation: Data is normalized, aggregated and generalized.

These have been implemented using Spark within Java. The processed dataset will be given to the other module as Spark data frames.

3.2.2 Statistical Analyzer

Statistical analyzer provides basic statistic of data set feed to the framework. It provides API to get Simple statistics like mean, variance median, column statistics to complex statistics like FP growth Algorithm based frequent item sets. The Statistical Analyzer module provides an API to perform basic statistical analysis on the crime dataset which is provided by the user. It enables user to get following analytical outcomes.

- Overall summary of dataset
- Column wise statistics: Mean, Variance, Median.
- Complex statistics : Frequent itemsets based on FP growth algorithm.
- Functions and function prams

3.2.3 Machine Learner and Predictive Analyser

In this platform, predictive analytics methods are mainly used for predicting category of a crime which can be occurred somewhere at a given time. In order to integrate predictive analytics features, it is necessary to have a machine learning component as well.

There are few possible approaches in order to integrate machine learning component with the platform. Since Apache Spark along with its Machine Learning Library R provides a rich facility for machine learning and data mining techniques, using Apache Spark R could be a better choice.

Using R within in the platform can be implemented in several different ways. One is using rJava library which provides facility to call and use R functions from java programs.Another one is using SparkR library which is provided by spark itself.A research done with a FBI data set has suggested Linear Regression as the best algorithm for predictions [23].

We have implemented the Machine Learning component which is built on top of Apache Spark MlLib and created in a way as to hide the complexity of spark Machine learning algorithms. Since crime data has lot of textual data it generates huge inconsistencies and errors when going to feed into ml component. This module basically provides following functionalities.

- Random Forest Classification
- Multilayer Perceptron Classification
- Decision Tree Classification

3.2.4 Descriptive Analyzer

Descriptive Analyzer use both quantitative and qualitative data and analytical techniques. Qualitative data and analytical techniques refer to non numerical data as well as the examination and interpretation of observations for the purpose of discovering underlying meanings and patterns of relationships. This is most typical of field research, content analysis, and historical research. Quantitative data are data primarily in numerical or categorical format. Quantitative analysis consists of manipulations of observations for the purpose of describing and explaining the phenomena that those observations reflect and is primarily statistical. Descriptive Crime analysis employs both types of data and techniques depending on the analytical and practical need. For example, crime data can be used in various ways, both quantitatively and qualitatively. The information such as date, time, location, and type of crime is quantitative in that statistics can be used to analyze these variables. Apart from spatial distribution of crimes, this considers the temporal relationship between the crimes, geographical situation, Population density and weather condition to give more detailed description about crime [24].

3.2.5 Prescriptive Analyzer

Urban areas are divided into several police districts usually based on demographic and geographic parameters. This is mainly done to divide the workload and to address the issues which includes homicides, burglaries and other kind of crimes more effectively by dividing the resources available for the police forces in a more efficient way. For example San Francisco is divided into 10 police districts [25]. Most of the police districts that exist today have been formed many years ago and with changes happening over time and movement of population we have to question whether the existing police districts provides for the purpose they were created [8].

As part of the solution Inquisitors Crime Data Analytic platform can redraw police districts boundaries with regard to the distribution of population in a more uniform manner [26].The platform provides capability to redistrict given country for any amount of possible police district according to the population distribution.

As second objective of sub system Prescriptive analyzer shown in 3.1, System can rationalize police patrol beats inside redrawn police districts.System takes Road network distance instead of traditional manhattan distance to calculate weighted crime distance between points. Also the system takes geographical constraints into account by going to the smallest geographical census level available.Inquisitor Crime Data Analytic platform uses heuristic based approach to rationalize police patrol beats in efficient way.For improve the functionality of police patrol beats and make them more effective,workload distribution among crimes will be equal and provide very low police patrol response time.

3.2.6 Application Programming Interface (API)

Each of the core modules Predictive analyzer,Prescriptive analyzer, Discriptive analyzer and the Preprocessor use facade Design pattern to hide the complexity of the of the module and provide necessary functionalities through simplified interface. Using Crime Data Analytics Framework API, it provides all the functionalities of the framework through an API. Both types of functionalities which include in the facades and not included in the facades can be gained through the Crime Data Analytics Framework API. For an example PreprocessorFacade in preprocessor module only provides the functionalities to handle missing data by deleting the entire row. But thorough the API, user can select whatever the missing value hadndling option implemented in the preprocessor module according to his/her preference (like predict the missing value of replace using the frequent item and so on).

3.2.7 Visuzlizer

The visualizer component enables the user to interactively analyze the dataset and the results provided by the predictive, prescriptive and statistical analyzer. This component provides data visualization through models such as histograms, heat maps and tables,pinmaps.The data required for the visualization is retrieved by connecting with the prescriptive, predictive and statistical analyzers through the provided API.

3.3 Design Documents

3.3.1 Use Case View

The crime data Analysis platform has design to handle specific use cases.Figure 3.2 shows use cases of the CADP.

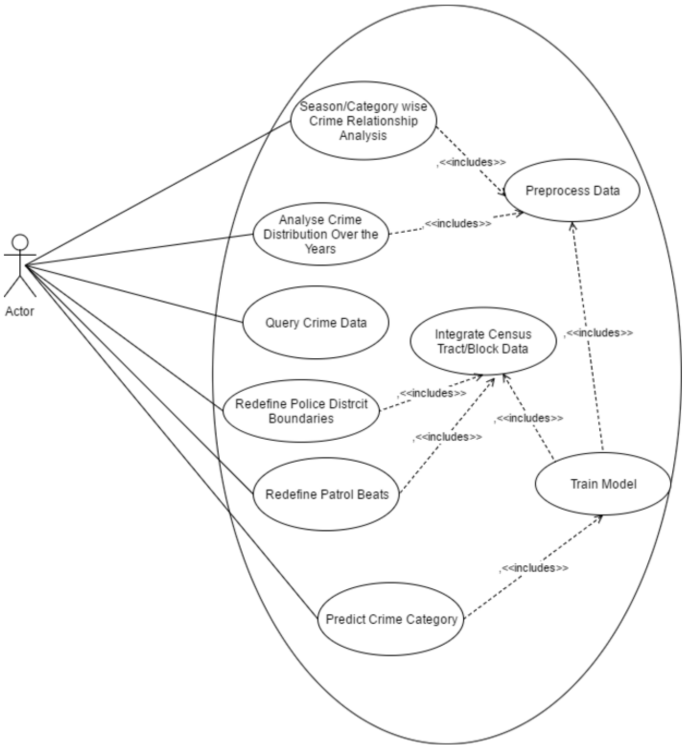


Figure 3.2: Use case diagram for CDAP

3.3.2 Development view

Figure 3.3explain the whole system in the developers perspective. There are six main components which are loosely coupled with each other

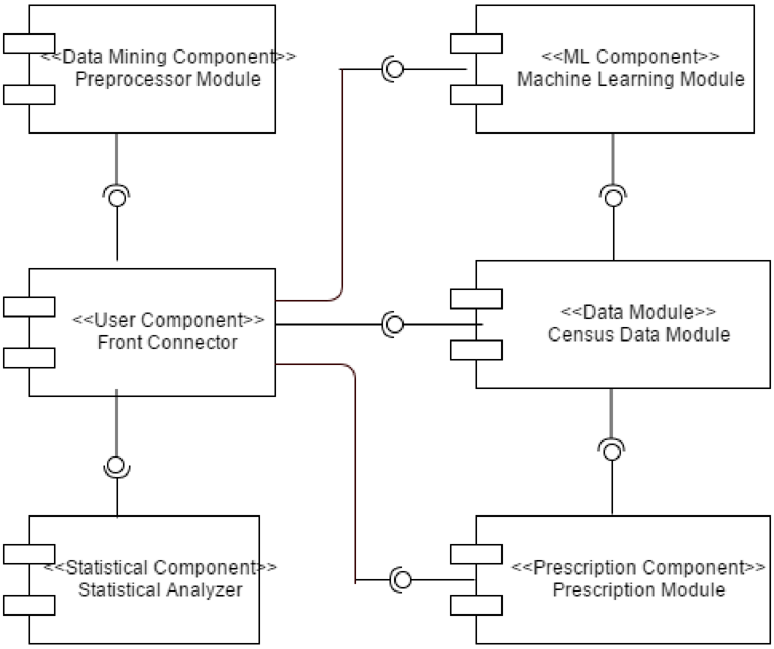


Figure 3.3: Use case diagram for CDAP

3.3.2.1 Preprocessor module

Preprocessor Module consists of the features and functionalities which are needed to preprocess the data before feeding them into other modules such as Machine Learner and Statistical Analyzer.Within this module, functionalities needed to preprocess the input data have been implemented.

3.3.2.2 Front connector module

Front connector provides api to the user by hiding the complexity of the Crime Data Analytic Framework. Users can feed data into the frame work and get results back using front connector module.

3.3.2.3 Statistical Analyzer module

Statistical analyzer provides basic statistic of data set feed to the framework. It provides API to get Simple statistics like mean, variance median, column statistics to complex statistics like FP growth Algorithm based frequent item sets. The Statistical

Analyzer module provides an API to perform basic statistical analysis on the crime dataset which is provided by the use.

3.3.2.4 Machine learning module

Machine Learning component is built on top of Apache Spark MlLib and created in a way as to hide the complexity of spark Machine learning algorithms.

3.3.2.5 Census Data module

This module keeps various kind of census data like census block data, census tract data race data

3.3.3 Data Description

This platform is based on Apache Spark engine. Therefore, within the framework inside the platform, data will be kept and processed as data structures used by Apache Spark such as Data Frame, Dataset. But since the user should not have to aware of how to use spark, those spark data structures have been hidden within some new custom data structures which enables user to use the platform easily without a knowledge about Apache Spark. Basic data flow within the platform is shown in Figure 3.4.

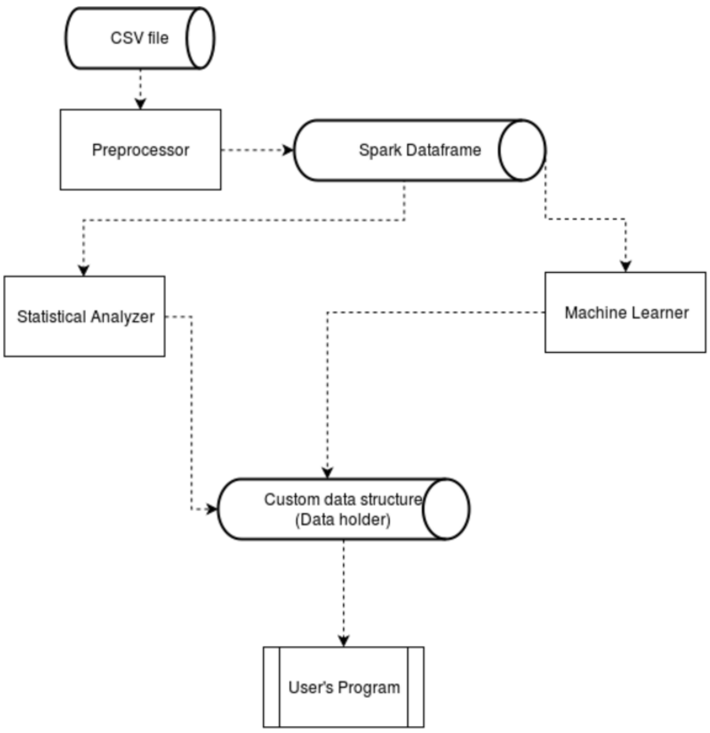


Figure 3.4: Data flow within CDAP

4. Implementation

We implemented a web application to visualize the project output to the user. This web application provides user the access to all the implemented features of the platform, including descriptive, predictive, and prescriptive analytics. Section 4.1 describes the development framework. Section 4.2 describes the implementation of redrawing of police district boundaries while Section 4.3 describes the implementation of police patrol beats generation. Section 4.4 describes the implementation of predictive analysis of crime category in the platform. Implemented features with screenshots from the web application are presented in Section 4.5.

4.1 Web Application

Crime Data Analytic Platform performs descriptive, prescriptive, and predictive analysis of crime data and provide user friendly and effective ways to analyze crime data. To make crime data analysis feasible and effective a web application has been developed to interact with the user. This has been developed using AngularJs along with Spring MVC.

Crime Data Analytics Platform provides visualization using a web application which has been built using Spring framework along with AngularJs. Reasons for selecting Spring framework for our platform development are as follows:

- Spring provides a very clean contrast between controllers, JavaBean models, and views.
- Spring is very flexible. It is entirely based on interfaces.

- Spring is view-agnostic. It doesnt push a developer to use JSP and can use alternative view technologies.

- Since we are using Apache spark java, we had to use a framework which is compatible with java platforms. Therefore we used Spring for the implementation of the web application.

Reasons to use AngularJs along with Spring framework.

- Directives in angularJs bring additional functionalities to HTML.

- Angularjs Material provides already developed UI components to be used.

- Provides two-way data-binding.

- Provides client side MVC framework.

- Dependency injection.

4.1.1 Features provided by the platform

1. Upload a crime dataset and population dataset.

2. Preprocess the uploaded crime dataset.

3. Discover spatial and temporal patterns of crime data using visualization features.

4. Query data in the crime dataset.

5. Redraw efficient police jurisdiction boundaries.

6. Draw efficient police patrol beats based on the crime distribution.

7. Predict crime categories for a given crime scenario.

4.2 Police District Boundary Redrawing

4.2.1 Overview

Major goal of police management is to find an equitable and efficient distribution of patrol resources across their city or jurisdiction. To make this distribution more manageable, police departments face the mammoth task of partitioning their jurisdiction into command districts or precincts. Each command district usually has a headquarters and commanding officer to oversee its police operations. Districts are further subdivided into patrol sectors or beats, with at least one patrol car assigned to each beat. On an even smaller scale, sectors are composed of reporting districts, the smallest geographical area for which police statistics are kept and reported to interested parties [27].

Better districting plans lead to lower response times, officers familiarization with their assigned area, more efficient use of personnel, more equal division of workload, a visible police presence, enhanced officer safety, officer accountability, and balanced police response to calls [28]. Traditionally, these geographic patrol boundaries are drawn by hand based on a police departments knowledge, experience, and the available police resources [11]. Most police departments also lack a method for formally evaluating and comparing the performance of competing district plans, instead relying on the judgement and intuition of police planners. However, given the complexities of the police districting plan, it is unlikely that an optimal districting plan will be chosen by chance using this method [29]. Moreover, as almost always these districts or zones are formed for some form of jurisdiction, the districts formed should be compact in space in order to facilitate the application of laws and regulations within the district [30].

In Inquisitor CDAP two methodologies are introduced to redefine command districts and patrol sectors or beats. One of the main objectives of CDAP is to redefine existing police district boundaries using existing crime and census data. This section focuses on the task of redrawing the police district boundaries of given district or the entire county. In this process heuristic-based clustering method has been used to divide population of the given region into equitable manner among the police districts and also to generate district in a compact manner. This process is important to provide stronger and more organized policing across the region. This section includes a computational case study using census data from the San Francisco County, USA which reveals the benefits of this approach.

4.2.2 Proposed Algorithm

Districting is the process of dividing a larger geographic space into smaller regions or districts or zones. In our heuristic-based clustering method,

1. Each district is represented by a cluster.
2. Cluster begins as a polygon(Or Seed Point)
3. Each cluster is grown by adding neighbouring polygons to it which are chosen based on heuristic value F.
4. Each polygon is a census tract.

Here the function value F is consists of,

1. Gap between target population and existing population of each cluster.
2. Compactness measurement (Isoperimetric Quotient).
3. Cost of growing the cluster (Number of neighbours of the cluster).
4. Each polygon is a census tract.

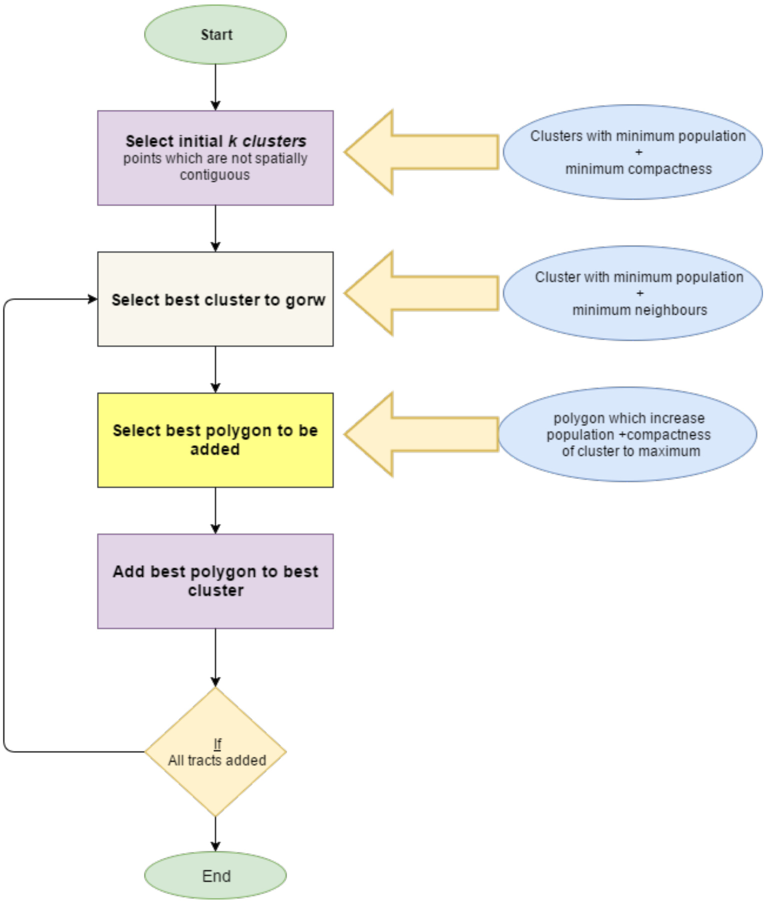

Start
Select initial *k clusters*
points which are not spatially contiguous
Clusters with minimum population
+
minimum compactness
Select best cluster to gorw
Cluster with minimum population
+
minimum neighbours
Select best polygon to be added
polygon which increase population +compactness of cluster to maximum
Add best polygon to best cluster
If
All tracts added
End

Figure 4.1: Population distribution of police districts

Figure 4.1 shows the major steps of heuristic-based clustering method, we have used for redrawing police district boundaries. In the first step the initial k clusters are selected. Here the initial k clusters are the census tracts which are having minimum population and minimum compactness. Apart from being that, each pair of census tracts should be at least apart from given distance. That is to ensure that selected census tracts are not spatially contiguous. Second step is to selct the best cluster to grow from the initially selected k clusters. It is done based on the the number of neighbours which each initial cluster has. So the cluster which is having least number of neighbours and the cluster which is furthest from the target population is selected as best cluster to grow. This is done because if a cluster other than the cluster with minimum number of neighbours is selected the available number of neighbours to the cluster which having minimum number of neighbours will be further reduced. Final step is to adding the best neighbouring polygon to the selected cluster. The polygon which takes the population of cluster closest to target population and which increases the compactness of cluster to the maximum is selected as best cluset. So this last two steps are iteratively carried out until all the polygons are added to k number of clusters. Each district is formed by clustering group of polygons that is spatially contiguous in a compact manner. Also, this clustering process is done based on optimizing target function which consists of population of each cluster and the compactness measurement. The police districts formed should be compact in space to facilitate the application of laws and regulations within the district. A compact district can be defined in many ways, but the best method is to use isoperimetric quotient.The Isoperimetric quotient of a plane figure is the ratio:

$$I = \frac{4 * \pi * A}{p^2} \tag{4.1}$$

where A is the area of the figure and P its perimeter. The Isoperimetric Inequality I states that this ratio is always less than or equal to one, with equality only for a circle. The shape is compared with the circle as circle is the most compact shape for any patrol beat because it covers the most area within the smallest perimeter [31].

4.2.2.1 Usage of Census Tracts

First part of the redistricting process is to extract data from three different sources to formulate the algorithm. Those sources contain the data about each Census Tract such as the population, mid point, and the coordinates of polygon which represents the Census Tract. Census Tracts are small, relatively stable geographic areas that usually have a population between 2,500 and 8,000.

A reason behind using Census Tracts as the building block of the cluster was that the Census Tract boundaries follow permanent and easily recognizable physical features. If physical features are not in close proximity or do not exist, street extensions, property lines, and municipal limits have been used as Census Tract boundaries. The advantage behind this is, if an arbitrary or random polygon has been used as a base of clustering, there can be scenarios where same property is divided among multiple districts. However, as the Census Tracts use existing boundaries above mentioned issues will not be occurred.

Another advantage of using Census Tract as base polygon is that the Census Tracts are defined in a such a way that it is as homogeneous as possible in terms of socioeconomic characteristics, such as similar economic status and social living conditions at the time of its creation. By clustering Census Tracts rather than clustering arbitrary polygons ensures that communities having similar socioeconomic characteristics, such as similar economic status and social living conditions, are not being divided among different police districts.

Another advantage is that Census Tracts are defined in a manner in which the shape of the Census Tract is as compact as possible. So that by clustering these compact polygon shapes rather than clustering arbitrary polygons help to maximize target function because target function evaluates the compactness of the cluster. Also, Census Tract boundaries respect census metropolitan area, census agglomeration, and provincial boundaries so while clustering Census Tracts the algorithm does not have to apply constraints to prevent clusters from violating provincial boundaries as well.

4.2.2.2 Defining Seed Points

Next step of the algorithm is to define seed points or the starting polygons of the clustering process. Number of seed points *k* are equal to the required number of police districts within the region. These *k* points are selected as the first *k* Census Tracts with maximum function value F where,

A - area of the cluster

p - perimeter of the cluster

G - Isoperimetric quotient

$$F = (X - Population) - G \tag{4.2}$$

$$G = \frac{4 * \pi * A}{p^2} \tag{4.3}$$

$$X = \frac{TotalPopulationOfCounty}{k} \tag{4.4}$$

Here apart from being the Census Tracts which are having least amount of population, they should also satisfy that they are minimally apart from given distance D.

$$D = \sqrt{\frac{AreaofMinimumBoundingRectangleCoveringpolicedistrict}{k * \pi}} \tag{4.5}$$

This is to ensure that seed points are not spatially contiguous. After selecting seed Census Tracts each one of these Census Tracts become a cluster and they are defined as base or seed clusters.

4.2.2.3 Selecting Best Cluster to Grow

In each iteration a cluster should be selected to grow and it is selected using the cost of reduction of flexibility to growth of cluster G.

$$MaxKs.t.K = (H - j * number\ of\ neighbours\ ) \tag{4.6}$$

By selecting the maximum K it ensures that the selected cluster is the one which will be hardest to grow. Also, by combining it with H and getting the maximum value the cluster which is hardest to grow with having minimum population can be selected to be grown. Here *j* is the coefficient use to take number of neighbours in to same range of H.

4.2.2.4 Selecting the Best Polygon to Add

After selecting the best cluster to grow calculate the value of function F, after adding each neighbour to the cluster. Where F is,

$$F = (X - \textit{ Population after adding polygon }) - j * G \tag{4.7}$$

By selecting the polygon which gives the least cost of F it guarantees that the selected polygon is the one which will (Here *j* is the coefficient use to take number of neighbours in to same range of H),

1. Take population of the cluster closest to the target population value
2. Have the minimum cost to grow.

4.2.3 Case Study

As our case study we applied our algorithm to redefine police district boundaries of San Francisco County. To measure the inequality of population distribution of newly created police districts we have used Gini Coefficient. The Gini coefficient (or Gini ratio) is a summary statistic of the Lorenz Curve and a measure of inequality in a population. The Gini coefficient ranges from a minimum value of zero, when all individuals are equal, to a theoretical maximum of one in an infinite population in which every individual except one has a size of zero [32]. The calculated Gini Coefficient for new population distribution was 0.02468 and it suggest that the population variance among districts is low. Figure 4.2 shows the distribution of population among police district boundaries for *k=10*. *k=10* is selected for the case study because originally San Fransisco is divided in to 10 districts.But it is possible to generate any number of *k* police districts according to the existing police resources.

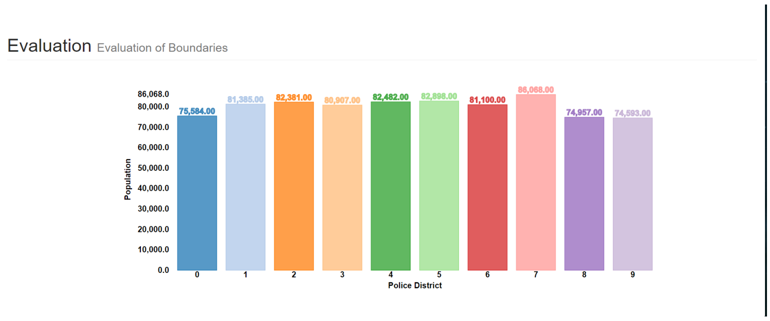


Figure 4.2: Population distribution of police districts

Also, as the compactness measure we have used Isoperimetric Quotient and we got a average value of 0.4957 for each district and which suggests the generated districts are compact. Because of this we can conclude that our algorithm was successful in the process of redistricting police district boundaries by generating more compact and equally distributed by population. The k value used to obtain the above results was 10. It was tested for k values which were less than 10 and values which were greater than 10 and it still gave satisfactory values for Gini Coefficient and Isoperimetric Quotient.

4.3 Police Patrol Beats Generation

4.3.1 Overview

This section focuses on division of a given police jurisdiction into equitable workload patrol beats and minimize 911 response time.It explores a way to find optimum patrol beats boundaries for a given police jurisdiction area based on the past crimes and their severity. Geometry is used to partition patrol beats appropriately. Based on the objective of the user the system can generate a customized, equality work divided police patrol beats for different seasons, for different time ranges such as night shift, early shift, and day shift patrol beats for weekdays and weekend days. More importantly system is capable of generating any amount of police patrols according to the users need. As an example, different police districts may need to allocate different number of police patrol beats according to their existing resources, geographical constraints, and as well as special occasions like elections.

4.3.2 Algorithms and Research Work

Patrol beats generation algorithm divides entire selected police district space into a given number of patrol beats with equitable workload. Algorithm runs at census block level.

Flow diagram of the algorithm used to rationalize police patrol boundaries is shown in Fig. 4.3. First, it takes the crime dataset and census data and tries to make a spatial

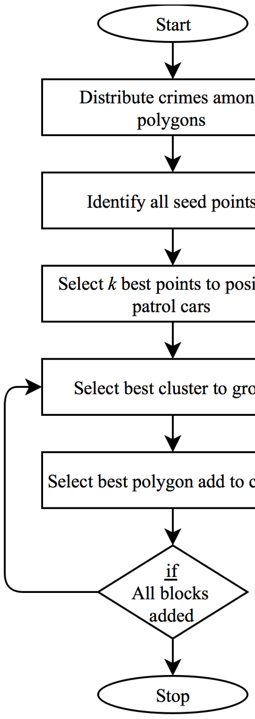

Start
Distribute crimes among polygons
Identify all seed points
Select *k* best points to position patrol cars
Select best cluster to grow
Select best polygon add to cluster
if
All blocks added
Stop

Figure 4.3: Flowchart of patrol beats generation algorithm.

relationship between them. For that the crime locations are distributed to appropriate polygons (i.e., census block) according to the geographic location of the crime. After this step, each polygon may have zero or more crime incident locations inside each of them. Then in the second step, we identify the optimal crime location which minimizes the total police workload inside each polygon. This optimal crime location is named as *optimal seed point*. After the second step algorithm generates N optimal seed points such that there is only one for each polygon. In the third step, it identifies the best k seed points to position the police patrol cars from the N seed points ($k \leq N$). Selected k points are not spatially contiguous. These k points are the initial positions of the clusters that are to be generated by aggregating the polygons. Step four, iteratively selects the cluster of polygons which has the current minimum workload to expand. Step five identifies the best neighbor polygon to add to the cluster using the heuristic value. The detailed description of each step with equations is presented next.

4.3.2.1 Distributing Crimes Among Polygons

According to their geographical location, crimes are mapped to their respective polygon. This spatial distribution of crimes will introduce high crime density in some polygons and low crime density in others. High crime density polygons may eventually lead to high police workload in those polygons and they may need some priority consideration while positioning the patrol beats.

If the patrol beat assignment is to be considered regardless of temporal variation of crimes, all crime locations of a given time period can be mapped to respective polygons. If temporal aspects such as time of day (daytime vs. night) or day of week (weekdays vs. weekends) to be considered, crime locations of different time periods need to be mapped to respective polygons separately. Then the next four steps need to be carried out on each dataset separately. This will result in a different set of patrol beats for different times of day or day of week.

4.3.2.2 Identifying All Seed Points Inside Polygons

Once all related polygons are selected, the best seed point inside each polygon is calculated. For that it uses actual road distance to calculate best possible seed point inside each polygon. They are selected in a way that minimizes the total workload. Weighted crime distance vector **V** is calculated for each polygon p as follows:

$$V_i = \sum d(i,j) * w(j) \tag{4.8}$$

where $\mathbf{V}_i$ is the i-th element of the weighted crime distance vector **V** of polygon p. $d(i,j)$ is the actual road distance from crime point i to crime point j and $w(j)$ is the weight of crime point j which indicates the severity of crime. For example, homicide has higher weight compared to a burglary. Table 4.1 lists all related symbols and definitions. From each calculated **V** which corresponds to each polygon, algorithm then identifies $Min(V_i)$ and treats the corresponding crime point i as the seed point in that polygon. This is because it is the optimal point which minimizes the police work in that selected polygon. In other words, that point has a high chance of being the

Table 4.1: Symbols used in equations.

Symbol	Description
$d(i,j)$	Actual road distance from crime point i to crime point j in polygon p.
D	Distance between cluster seed point and selected neighbor polygon seed point.
G	Define clusters ability to grow itself, after polygon p is added.
H	Defines the fitness of the cluster after adding polygon p.
k	required number of police patrol beats.
L	Cluster compactness after new polygon p added.
$RA(k)$	Minimum rectangular area need to cover cluster when cluster having k polygons.
$RA(k+1)$	Minimum rectangular area need to cover Cluster after adding polygon p with $k+1$ polygons.
$\mathbf{V}_i$	Weighted crime distance of i-th crime point in polygon p.
$w(j)$	Weight of crime point j (based on the severity of crime).

highest crime occurring location in that polygon.

4.3.2.3 Positioning Police Cars

From the calculated seed points we select k seed points where k is the required number of police patrol beats in the police district ($k \leq N$). These k points are selected from the first k polygon seed points with the highest weighted crime distance. Here, apart from being the polygons which have the highest weighted crime distance, they should also satisfy that they are minimally apart from given distance D where:

$$D = \sqrt{\frac{A}{no\ of patrols * \pi}} \tag{4.9}$$

A is the area of minimum bounding rectangle covering police district. Calculated D value ensures that the patrol car positioning seed points are not spatially contiguous. After selecting positioning points, those seed points will be grown as a clusters of

Algorithm 1: Draw Patrol Beats Boundaries

```
inputs : Initial seed points $S$
output: Final patrol beasts boundaries $D^*$
$followersList$ ←n-S;
while $followersList.size \neq 0$ do
    $c$ ←selectBestClusterToGrow;
    $neigbours$ ←getAllNeighbours(c) ;
    $fitness \leftarrow \emptyset$;
    while $neigbours.size \neq 0$ do
        $f$ ←getFollower;
        $L$ ←calcArea(c) - calcArea($c + f$) ;
        $D$ ←calcRoadDistance($c + f$) ;
        $G$ ←calcGrowthFactor($c + f$) ;
        $fitenss$ ←G + $D-$L ;
    end
    $c$ ←c+neighbours.get(fitness.getMax()) ;
    $D^*$ ←c ;
end
```

polygons later. The selected police car positioning points are defined as leader or seed of patrol beats. Other remaining $n-k$ become followers.

4.3.2.4 Generating Patrol Beats

Initially each k cluster have one polygon each, and iteratively more polygons are be aggregated to their clusters. Algorithm 1 illustrates how the polygons are aggregated. The clusters are first sorted according to their total cluster workload. Then in each iteration, cluster with the minimum workload gets selected as the growing cluster, because to obtain equitable workload it is essential to give priority to the cluster which has smallest workload. Each cluster has a list of neighboring polygons which are not yet aggregated to the cluster. Using heuristic in equation 4.10, it selects the best neighbor polygon which can be added to the cluster in each iteration and iteratively grow the cluster until the termination point where all the polygons have a cluster.

$$H = G + D - L \tag{4.10}$$

where H is the heuristic value that defines the fitness of the cluster after adding free neighbor follower polygon p to cluster. Growth factor G represents the cluster's ability to grow itself after the block is added to the cluster. More neighbors in a cluster will lead to higher growth factor.

Distance factor D can be calculated as follows:

$$D = \frac{C}{Road\ distance\ between\ two\ seed\ points + 1} \tag{4.11}$$

Actual distance between cluster seed point and selected neighbor polygon seed point is measured using Open Street Map data. Distance is calculated using GraphHopper which is a fast and memory efficient Java road routing engine that uses Open Street Map data [33].

It is essential to select these two points close to each other. If not, cluster will be more lengthy and unnecessary traveling costs will be added to the cluster. Low actual road distance between two seed points provides higher value to the heuristic function. C in the equation is a constant.

Lengthiness factor L indicates what would happen to the cluster compactness after new polygon p is added. It gives an insight about the clusters growth direction. Lengthiness gives a negative effect to the heuristic function because patrol beats must be more compact. If the cluster is growing in a very disoriented manner, lengthiness factor produces a large value and eventually it makes low H value.

$$L = RA(k) - RA(k+1) \tag{4.12}$$

Algorithm adds fitness value of each neighbor into a list and selects the fittest neighbor to add to the cluster. Algorithm runs until all the selected n polygons are assigned to clusters. Finally, each cluster is taken as a separate police patrol beat.

4.3.2.5 Usage of Census Block Data

Census blocks are statistical areas bounded by visible features such as roads, streams, and railroad tracks, and by nonvisible boundaries such as property lines, city, township, school district, county limits, and short line-of-sight extensions of roads. It is the building block for all geographic boundaries that census bureau tabulates as data, such as tracts, places, and American Indian Reservations.

The crime data set we are using as case study is sample of Californian crime data. To obtain block details, we gathered census block data from US Census Bureau. Block data is integrated into crime data set to get an understanding on how crimes are distributed according to the census blocks. Then according to the crime distribution of each block (by considering severity of the crimes and minimum road travel distance police patrol wants to travel to locate one place to another), it is possible to find seed points inside each census block which can cover all existing crime points in each block with a minimum police patrol work. The minimum work which is calculated, is assigned to each block for representing the blocks total police work. Using that seed point in each block, algorithm can identify best possible seed points for patrol beats generation based on the police requirement.

This patrol beat is designed for the redesigned police districts which was discussed in Section 4.2. The system can identify the tracts and blocks related to the given police district and using those census tract and census block data it generates police patrol beats for the respective police district.

4.3.2.6 Measure Distance Between Crime Locations Along the Roads

Actual road distance is taken into account for all cases when measuring distance. It is measured using openstreetmap using graphhopper API. This is because measuring traditional straight line distance does not give context sensitive and precise information about travelling distances. Also, it does not provide how the city road network is structured, traffic conditions, and geographical features like rivers, parks, bridges which can slow down or increase the patrol car travelling time.

4.3.3 Case Study and Evaluation

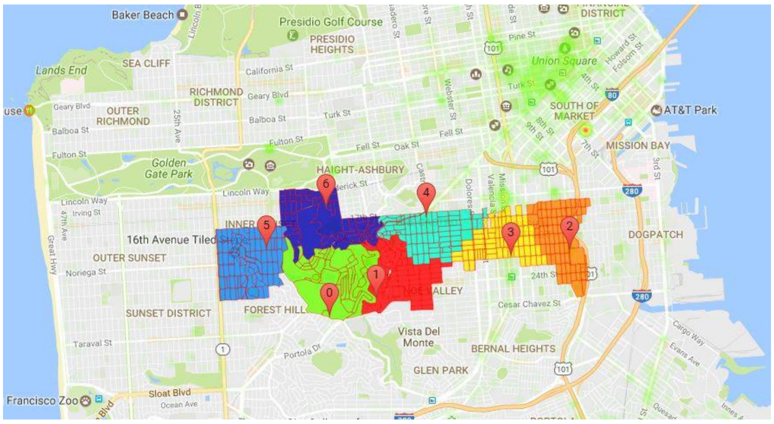


Figure 4.4: Police district boundary generation for one police district

Police patrol district design presents a multi-objective optimization problem with two goals; namely minimizing workload variation between patrol districts and minimizing the response time for officers responding to calls for service. As a case study, we applied our algorithm to rationalize police patrol beats of San Francisco County, CA, USA. We selected crime data from 2003 to 2015, which was downloaded from SF open data [34] and consists of nine attributes such as timestamp, category, description, and location (Latitude and Longitude). To obtain block details, we gathered census block data from US Census Bureau. Block data is integrated into crime data set to get an understanding on how crimes are distributed according to the census blocks. We used San Fransisco Open Street Map to measure the actual road distance between points and use census blocks as minimum building block of the polygons. We implemented the proposed algorithm on our Inquisitor crime data analytic platform.

We considered equitable workload, patrol car response time, and compactness of the beat while evaluating the algorithm. To measure the inequality of crime distribution of newly created police patrol beats we used Gini Coefficient. The Gini coefficient (or Gini ratio) is a summary statistic of the Lorenz Curve and a measure of inequality in workload distribution. The Gini coefficient ranges from a minimum value of zero, when all individuals are equal, to a theoretical maximum of one, in an infinite population in which every individual except one has a size of zero [35].

Districting problems of every type has a concern about geographical compactness to measure the effectiveness of the designed district. A bigger or a longer cluster will have a longer emergency response time. It is understandable that if a specific patrol beat does not have a good compactness measurement, it will definitely affect the response time to fast response to CFS within that patrol beat. Therefore, for our algorithm we considered compactness as another factor for evaluating police patrol beats. As the compactness measure we used Isoperimetric Quotient. The Isoperimetric quotient of a plane figure is the ratio:

$$I = \frac{4 * \pi * A}{p^2} \tag{4.13}$$

where A is the area of the figure and P its perimeter. The Isoperimetric Inequality I states that this ratio is always less than or equal to one, with equality only for a circle. The shape is compared with the circle as circle is the most compact shape for any patrol beat because it covers the most area within the smallest perimeter [31].

Fast response to Calls for Service (CFS) of citizens improves the chances of arresting offenders, increases the chances of identifying and locating witnesses, provides immediate gathering of physical evidence, provides immediate life-saving aid, enhances the reputation of the police department, and increases citizen satisfaction with police [36]. Therefore, it is important to consider response time as a factor when evaluating the effectiveness of designed police patrol beats.

4.3.3.1 Equitable Workload

Fig. 4.4 illustrates the police patrol beats generated by the algorithm and initial patrol car positioning points. Workload is the main criteria for patrol beats evaluation. The variation of workload between districts could arise because crimes tend to cluster in specific hotspots rather than being distributed uniformly across a city or a given area. If the workload variation between districts is high, some officers belonging to patrol beats might experience much higher number of CFS. Therefore, it is important to consider workload distribution when evaluating the effectiveness of police patrol beats [36].

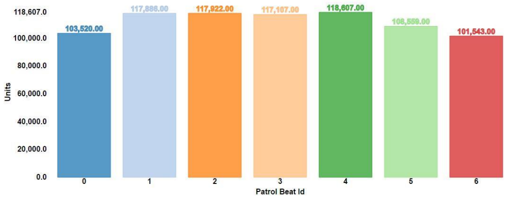


Figure 4.5: Workload distribution among patrol beats.

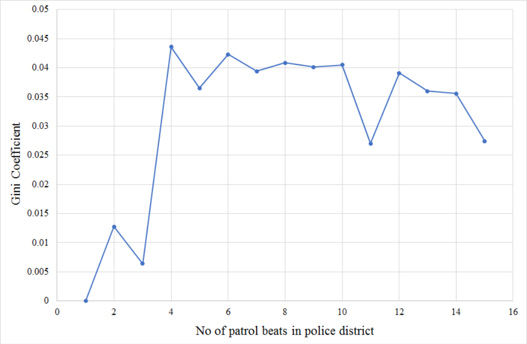


Figure 4.6: No of patrol beats in police districts vs. Gini coefficient.

Each patrol beat should have roughly similar amount of workload. The workload should be based on the severity of the crime and time taken to locate and deal with the crime. A single beatś workload can be calculated as sum of each census blockś weighted crime distance and distance to travel between blocks.

Fig. 4.5 shows the assigned workload for each patrol beat. Histograms with similar height indicate that workload is equally distributed among the generated seven patrol beats in the case study. Fig. 4.6 describes how the Gini coefficient is changed with the number of patrol beats inside a police district. Low average Gini index value of 0.036 proves that the workload is equally distributed among patrol beats.

4.3.3.2 Compactness of Beats

Compactness of the beat is evaluated using standard Isoperimetric quotient. Good compactness will give higher Isoperimetric quotient value. Fig. 4.7 shows the compactness of generated seven police patrol beats in Fig. 4.4. Fig. 4.8 shows how the average compactness of police patrol beats change with the number of police patrol beats inside police district. When the number of patrol beats increased, the isoperimetric quotient value converged to 0.7 indicating good compactness with in the

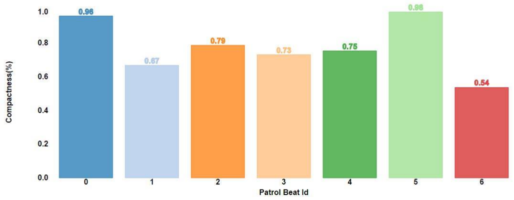


Figure 4.7: Compactness of each patrol beat.

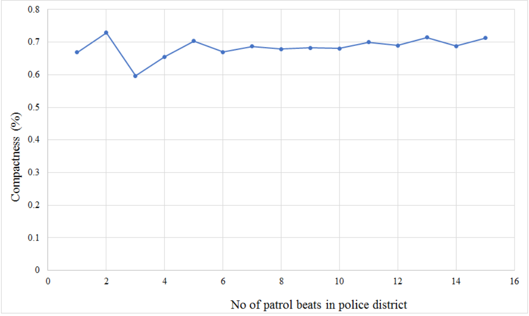


Figure 4.8: Average compactness of patrol beats.

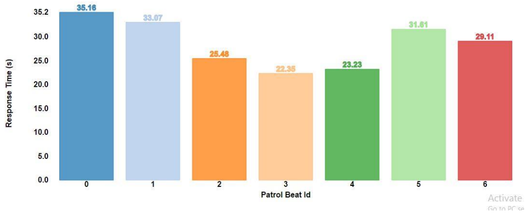


Figure 4.9: Response time in each patrol beat.

generated patrol beats.

4.3.3.3 Patrol Car Response Time

Fig. 4.9 shows the average time taken to travel to any point in the patrol beat from the seed point, where the patrol car is located. It must be clarified that geographical constraints such as rivers may introduce latencies to patrol beats. But the car position location is calculated in advance in a way that minimizes those impacts by taking those geographical constraints into account. According to Fig. 4.9, it shows that police car response time is very low. That means patrol car can locate any point inside the clusters generated with an average delay of 28 seconds. It is important to understand that this 28 seconds is the latency of patrol car to reach location. Therefore, the actual 911 response time will be 28 seconds plus other overheads. Fig. 4.10 shows how the

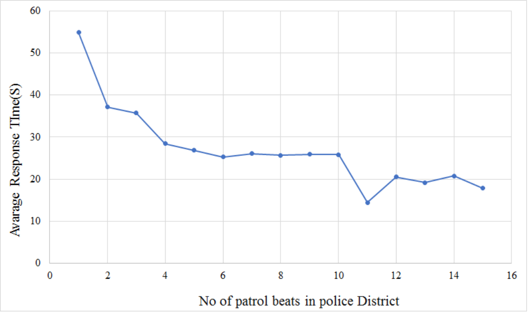


Figure 4.10: Average patrol car response time.

average response time is changed according to different number of police patrol beats. According to the graph, we can observe that when higher the number of police beats located inside police district, lower the time taken for the police car to reach the crime point.

4.4 Predict Crime Category for given Crime Scenario

4.4.1 Overview

Many law enforcing organizations around the world collect large amount of data daily. Analyzing this data can be very interesting to see what kind of patterns they would produce. This data can be used for many predictive tasks using Machine Learning (ML). task Predicting crime category (i.e., type of crime) that could occur in a given place at a certain day is one of the useful predictions. By predicting the highly likely targets to be attacked during a specific period of time and specific geographical location, police will be able to identify better ways to deploy the limited resources and also find and fix the problems leading to crimes. So to find a way to reduce crimes we look into ways of using ML to predict about crime types/category that are most likely to happen during a given time and location.

4.4.2 Dataset

We selected a San Francisco crime data from 2003 to 2015 to demonstrate the crime category prediction. The dataset was downloaded from SF open data [34] and consists of following nine attributes:

1. Date - timestamp of the crime incident (TimeStamp)
2. Category - category of the crime incident (String)
3. Descript - description of the incident (String)
4. DayOfWeek - day of the week of the incident (String)
5. Pd District - District of Police Department which the incident occurred (String)
6. Resolution - how the incident was resolved (String)
7. Address - approximate street address of the incident (String)
8. X - Longitude (double)
9. Y - Latitude (double)

Criminology theories states that the population also have a significant effect on crimes []. Therefore, block- wise population of San Francisco in 2010 was aggregate with the crime data set. The data set was collected from United States Census Bureau web site [37].The population dataset has the following attributes:

1. STATEF - State ID (Integer)
2. COUNTYFP - County ID (Integer)
3. TRACTCE - Census tract ID (Integer)
4. BLKGRPCE - Census Block ID (Integer)
5. POPULATION - Population of respective Block (Integer)
6. LATITUDE - Latitude (double)
7. LONGITUDE - Longitude (double)

After combining data from the above two datasets an aggregated dataset with the following attributes was created:

1. Year -(integer)
2. Month- (integer)
3. Day -(integer)
4. Hour- (integer)
5. Category - (String)
6. DayOfWeek -(String)
7. Police District - (String)
8. Resolution - (String)
9. Longitude (double)
10. Latitude (double)
11. Population - (integer)

4.4.3 Feature Selection

Dates, Category, DayofWeek, Resolution, Population, Latitudes, and Longitudes are the most relevant columns in the dataset. From the Date column we can extract year, month of the year, and hour of day. By using regular expressions we selected the time of the day at which the crime occurred using Dates attribute. Thus, we can obtain the time of crime as integer values between 1-24. Police districts are categorical values with string labels. There are other categorical values such as Dayoftheweek and Resolution. Using string indexer those string valued features were transformed into double values.

4.4.4 Feature Transformation

- Feature Standardization

An attribute is normalized by scaling its values so that they fall within a small specified range such as 0.0 to 1.0. It is essential to normalize data before feeding them into Nave Bayesian, Random forest, and Multilayer perceptron classification algorithms we implemented. Data normalization in project helped to prevent attributes with initially large values such as X and Y coordinates from outweighing attributes with initially small ranges (e.g., time of Day). Using rescale method in Data Transformation class in Apache Spark, we normalized the whole feature set to be fed into the classification algorithms.

- String Indexing

  String indexing encodes a string column of labels to a column of label indices. The indices are in [0, numLabels), ordered by label frequencies, so the most frequent label gets index 0 and least frequent get highest value. If the input column is numeric, we cast it to string and index the string values [38].

  DayOfTheWeek have seven indexes with Wednesday the most frequent one as index 0,pd district have indexes with SOUTHERN most frequent one as index 0 and so on.

4.4.5 Algorithms Used

Our first step in the solution approach was to pre-process the data attributes to match the solution model. Under preprocessing, initially we analyzed the data attributes. Thus, we learnt that rather than blindly using the training data for the prediction model, we must pre-process the data by removing inconsistencies and improving their usability. In addition, if a data record gives more prominent weightage for the result we should enhance its effect more than others. Also, if the result is independent of an attribute we should remove or reduce the weight or emphasize of that attribute.

Next, we had to construct a model for prediction. There, we had to select some suitable classification algorithms for our implementation. Instead of implementing those algorithms from the scratch, we use Apache Spark machine learning library [39], already available. The reason was Apache Spark provides a rich facility to easily work with big data along with ML learning and data mining capabilities. Because the memory utilization is managed well in Apache Spark, we did not have to concern much about points like allocating memory in our computers, processing power for the project

For our solution, models were trained using the Random Forest Classifier, Multilayer Perceptron Network, and Naive Bayes method. The accuracy of results from classification or prediction model depended on suitability of selected model and parameter values used with the model. The accuracy and the performance a model is affected by several factors including noise in the dataset, number of training data examples, types of attributes, dimensions and the constructed model.

When training the model using Perceptron Network, we used one hidden layer with 500 perceptron. We were not able to increase the number of hidden layers above 500 due to high computational time to construct the model

Finally, when look at the results we got from the three models we trained, we could notify that the model which is trained by Random Forest classifier product crimes categories with relatively has given results with the highest accuracy.

In ML, classification can be defined as the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data which is containing observations whose category membership is known. There are two categories in classification.

- Binary classification : In this method, only two classes are used
- Multiclass classification : In this method, an object is assigned to one of several classes.

To predict crime categories multiclass classification is needed because crime categories have more than two labels. While there are different machine learning algorithms for classification we used Naive Bayesian Classifier, Multi-Layer Perceptron Network, and Random Forest Classifier.

1. Nave Bayesian

   Nave Bayes learner or nave bayes classifier is a highly practical bayesian learning method. It is quite popular in ML applications, due to its simplicity in allowing each attribute to contribute towards the final decision equally and independently from the other attributes. This is because nave bayesian classifier assumes that attributes have independent distributions. Although this assumption is not always true, this algorithm has shown remarkable success in practice, often competing with much more sophisticated techniques. It uses the Bayes theorem which says,

   $$p(c\_j|d) = \frac{p(d|c\_j)p(c\_j)}{p(d)} \tag{4.14}$$

   Here, p(c_j—d) = probability of instance d being in class c_j , p(d—c_j) = probability of generating d given class c_j, p(c_j) = probability of occurrence of class c_j and p(d) = probability of instance d occurring.

   Because of the assumption that attribute have independent distributes, p(d—c_j) = p(d_1c_j) * p(d_2—c_j) * ... * p(d_n—c_j) [40]

   Here, p(d—c_j) = probability of class c_j generating instance d, p(d_1—c_j) = probability of class c_j generating observed value for feature 1, and so on.

   Nave bayes classifier are easy to implement and it also provides good results in most of the cases. Also nave bayes learners and classifiers can be extremely fast compared to more sophisticated methods. This efficiency is mainly due to the simplicity of Nave Bayesian classifier. This was implemented using Apache Spark mllib. Naive Bayes in mllib implements multinomial nave bayes.

2. Random Forest Classification

Random forests are one of the most successful machine learning models for classification and regression. [41].

Random forests algorithm does not over fit. It is achieved by combining many decisions trees. One can run as many trees as needed. Random forests algorithm is considered fast. Basic algorithm of Random forests trains a set of decision trees separately, so the training can be done in parallel. The algorithm injects randomness into the training process to make each decision tree different from each other. By combining predictions from each tree reduces the variance of predictions, improving the performance on test data. Following are some of the features of Random forests classification. Random forests handle categorical features and can even extend to multi class classification. Moreover, it is able to capture non-linearity and feature interactions. This algorithm runs efficiently on large datasets and can handle thousands of variables without variable deletion. Random Forest algorithm is effective for estimating missing data and maintains accuracy even when large parts of data are missing . [41,42]

For the model, we configured our random forest for 500 trees.

3. Multi-Layer Perceptron Network (MLPNs)

   Multi-Layer Perceptron (MLPs) are classified as a type of Artificial Neural Network. The computation is performed using set of (many) simple units with weighted connections between them. Multi-layer perceptron consists of some layers.

   They are namely,

   - Input layer: This is the bottom most layer that takes input from a dataset. It is the exposed part of the network.
   - Hidden layers: One or more layers which are not directly exposed to the input.
   - Output layer: takes the output from the final hidden layer. It outputs a value or vector of values that correspond to the format required for the problem. [19].

We used Apache Spark ML library for this implementation.

Before feeding data into above three ml algorithms we did the following feature transformations to the dataset things using Apache Spark.

Since crime data consist of string labeled data, it was needed to index those string labeled data. For that we used String Indexer, which is a feature transformation technique provided by Apache spark.

After that data set was indexed, Data set was standardized by scaling in a way such that values fall within a small specified range in 0.0 to 1.0.The standardization done using standard deviation and the mean. For Standardization we used Apache Standard Scaler class.

Then Vector Indexer of Spark was used to decide what categorical features are. Categorical values were transformed to their indices. Then transformed data (Spark Labeled Points) could be passed to Random Forest Classification algorithm.

4.4.6 Results and Evaluation

Since this is a multi-class classification problem, there is a tendency to, performance to be evaluated by computing just the accuracy of each model which is the proportion or percentage of correctly predicted labels over all predictions. However, we can always compute precision and recall for each class label and analyze the individual performance on class labels or average the values to get the overall precision and recall since accuracy alone is sometimes quite misleading. There may be a model with relatively 'high' accuracy with the model predicting the *not so important* class labels fairly accurate but the model may be making lot of mistakes on the classes that are actually critical to the application.

Accuracy : Overall correctness of the model and is calculated as the sum of correct classifications divided by the total number of classifications.

Precision : Given all the predicted labels (as given class X), how many instances were correctly predicted.

Recall (Sensitivity) : For all instances that should have labeled X, how many of these were correctly captured.

Table 4.2 shows the confusion matrix Of Naive Bayesian classification. Confusion matrix for Random Forest with 500 trees is shown in Table 4.3 while confusion matrix for Multilayer Perceptron neural network with 500 perceptron is shown in Table 4.4.

4.4.6.1 Confusion Matrix of Naive Bayesian classification

Table 4.2: Confusion Matrix of Naive Bayesian classification.

	Other offences (Actual)	Larcany/ Thefft (Actual)	Vehicle Thefts (Actual)	Non criminal (Actual)	Drug narcotic (Actual)
Other offences (Predicted)	56232.0	24415.0	0	0	361.0
Larcany/ Thefft (Predicted)	12769.0	39019.0	0	0	483.0
Vehicle Thefts (Predicted)	11093.0	16418.0	0	0	299.0
Non criminal (Predicted)	12388.0	10571.0	0	0	95.0
Drug narcotic (Predicted)	14053.0	1998.0	0	0	9.0

4.4.6.2 Confusion Matrix of Random Forest classification

Table 4.3: Confusion Matrix of Random Forest classification.

	Other offences (Actual)	Larcany/ Thefft (Actual)	Vehicle Thefts (Actual)	Non criminal (Actual)	Drug narcotic (Actual)
Other offences (Predicted)	38423	13637	280	1840	19
Larcany/ Thefft (Predicted)	8138	26624	75	54	8
Vehicle Thefts (Predicted)	4913	10686	25554	89	46
Non Criminal (Predicted)	8581	6247	124	362	3
Drug narcotic (Predicted)	7270	714	12	2671	0

4.4.6.3 Confusion Matrix of Multilayer Perceptron classification

Table 4.4: Confusion Matrix of Multilayer Perceptron classification.

	Other offences (Actual)	Larcany/ Thefft (Actual)	Vehicle Thefts (Actual)	Non criminal (Actual)	Drug narcotic (Actual)
Other offences (Predicted)	15312	63627	1261	1413	0
Larcany/ Thefft (Predicted)	25008	113232	257	730	0
Vehicle Thefts (Predicted)	25206	47611	283	449	0
Non criminal (Predicted)	32580	28107	295	494	0
Drug narcotic (Predicted)	40047	3164	34	58	0

Table 4.5: Accuracy recall and precision comparison.

Evaluation Type	Naive Bayesian	Random Forest	Neural Network
Accuracy	0.48	0.4	0.32
Precision OTHER OFFENSES	0.42	0.44	0.44
Recall OTHER OFFENSES	0.69	0.71	0.69
Precision LARCENY/THEFT	0.31	0.34	0.32
Recall LARCENY/THEFT	0.75	0.76	0.81
Precision VEHICLE THEFT	0.09	0.42	0.00
Recall VEHICLE THEFT	0.01	0.01	0.00
Precision NON-CRIMINAL	0.00	0.72	0.11
Recall NON-CRIMINAL	0.00	0.14	0.00
Precision DRUG/NARCOTIC	0.00	0.44	0.13
Recall DRUG/NARCOTIC	0.00	0.25	0.01

In Table 4.5 below graph the precision and recall values are compared. If we consider the accuracy of Nave Bayesian and Random Forest algorithms, the accuracy of Nave Bayesian (0.48) is higher than accuracy of Random Forest (0.40). Hence, if the algorithm is selected only by considering the accuracy the model that should have been selected was the Nave Bayesian algorithm. But that is misleading. If we consider the class labels such as NON-CRIMINAL, DRUG/NARCOTIC, and LARCENY/THEFT the precision and recall given by applying Nave Bayesian is 0. It means,

1. Precision: Out of the times label X was predicted, 0% of the time the system was correct.

2. Recall: Out of all the times label X should have been predicted only 0% of the labels were correctly predicted.

Therefore, if the Nave Bayesian algorithm is selected only by looking at accuracy, the possibility of predicting important class labels such as DRUG/NARCOTIC and LARCENY/THEFT would be 0. That would be a not suitable for our application. Because whenever our application predicts a theft, it would become a wrong prediction. Therefore, despite the higher accuracy we have chosen Random Forest algorithm considering the higher precision and recall values of individual labels. If we consider Random Forest algorithm and Neural Network algorithm the accuracy of Random Forest (0.40) is higher than accuracy of Neural Network algorithm (0.32) and also precision and recall of individual labels of Random Forest are also higher than Neural Network.We have chosen Random Forest algorithm considering the higher accuracy, higher precision and higher recall values of individual labels.

4.5 Crime Data Analysis Tool

This section provides screen shots taken from the web application.Figure 4.11 shows the general interface from which user has to navigate around the web application.

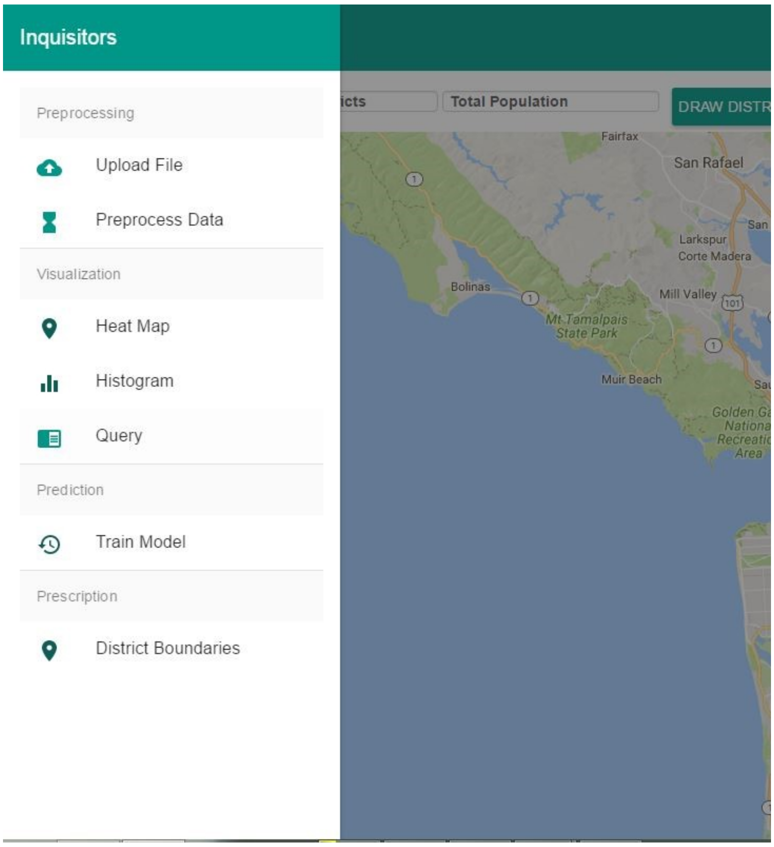
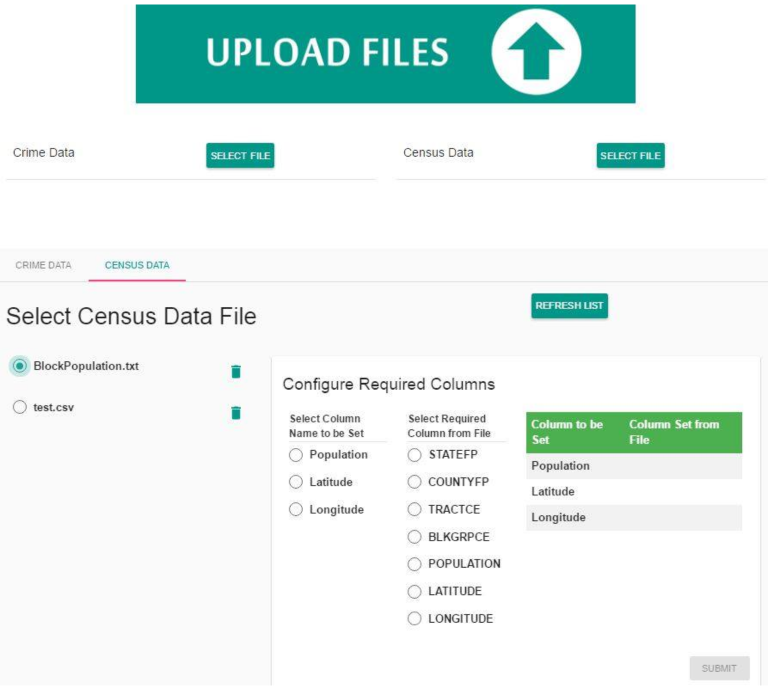


Figure 4.11: Web application interface with side menu.

4.5.1 Preprocessing

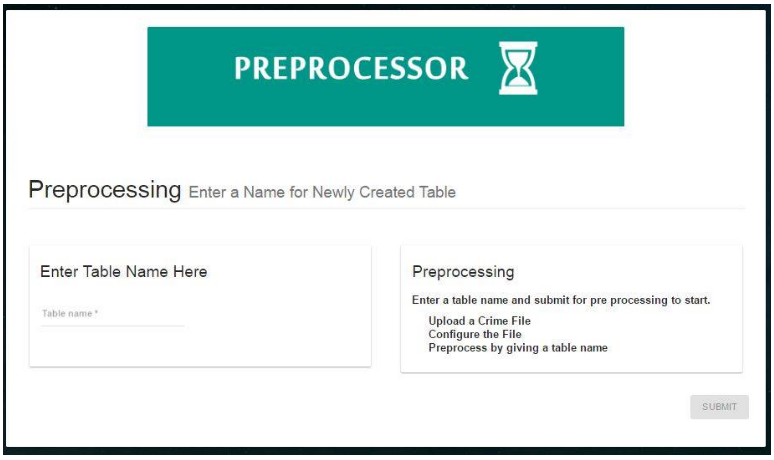
User gets the option of uploading the required crime data files and also population data files separately as shown in Figure 4.12. After uploading the files will be shown on separate lists (On separate Tab windows) for crime and census data. Once the file is uploaded a user can select a file from the list and then configure the required columns by selecting the column names which are shown on the interface. This is required to be done before moving on to other steps.



Column to be Set	Column Set from File
Population	
Latitude	
Longitude	

Figure 4.12: Interface to upload dataset files

Preprocess Data page provides an option of adding a name to the table created by preprocessing steps (see Figure 4.13). This comes in handy when a user needs to query through data in the table. This is also a required step to be carried out to before moving on to other steps.



Enter a table name and submit for pre processing to start.

Upload a Crime File
Configure the File
Preprocess by giving a table name

Figure 4.13: Web application interface for preprocessing data

4.5.2 Visualization

Visualization consists of three main options, namely Heat Maps, Histograms, and Query.

4.5.2.1 Heat Map

Heat Map page provides user with several options. Figure 4.12 shows the interface provided for these features:

- Period Wise

  It provides users to view crime data on a map by categorizing data specifically according to crimes that happened during the weekend, crimes that happened during the weekdays and also crimes that happened during the New Years Eve. This gives the user to visualize data and identify patterns of crimes with regard to different periods.

- Category Wise

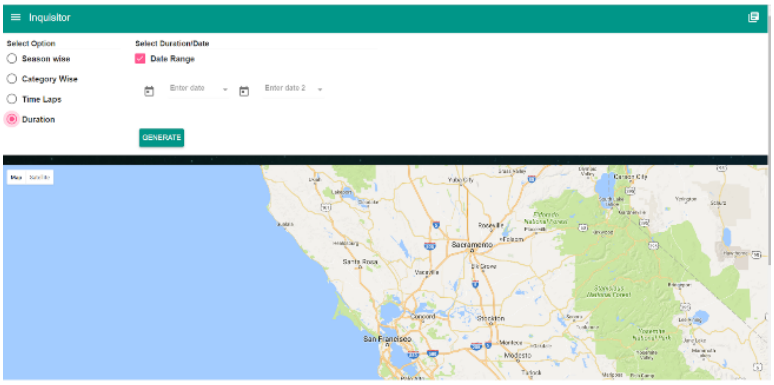
  Category wise provides the option for the user to view crime data according to categories which are provided in the data set.

- Time Lapse

  This option gives the user to generate crime data details on the heat map year wise which will increment when the play button is clicked.

- Duration

  This option provides the user with the ability to observe crime data within a specific range of days or within one specific day.



Select Option

○ Season wise
○ Category Wise
○ Time Laps
● Duration

Select Duration/Date

☑ Date Range

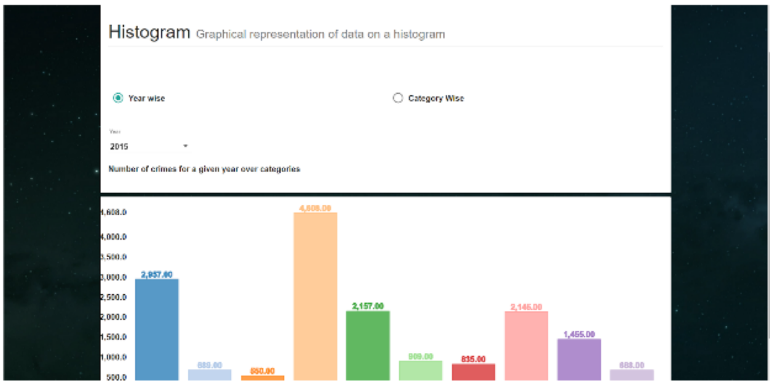
Enter date
Enter date 2

GENERATE

Figure 4.14: Web application interface for heat-map generation.

4.5.2.2 Histogram

This page offers the user to visualize crime data in yet another way, using graphs as shown in Figure 4.15. This provides the ability for the user to view data year-wise and also category wise. When user selects the option of viewing year wise, data are displayed for different categories of crimes related to that year. If a user selects the option of viewing data category wise, data are shown for different category for different years.



Histogram Graphical representation of data on a histogram

● Year wise ○ Category Wise

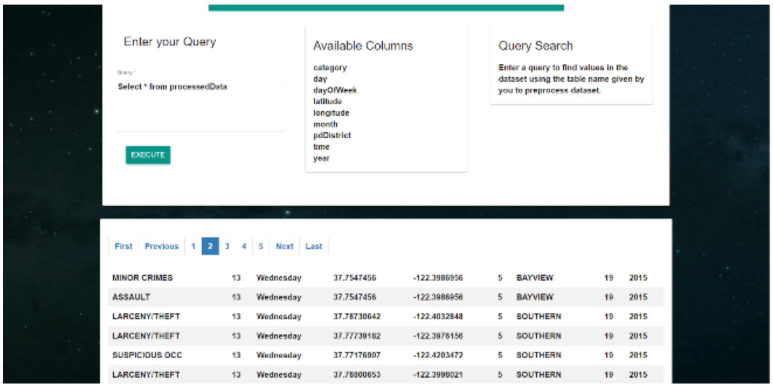
Year
2015

Number of crimes for a given year over categories

Figure 4.15: Web application interface to show histogram data

4.5.2.3 Query

Query page provides the option of querying through the dataset provided by the user. A user can enter SQL queries and execute them to get results. The intention of this page is to provide the user with a clear and easy way of going through the available data in the dataset and also recognize different patterns. Figure 4.16 shows the web application interface.



Enter your Query

Query *
Select * from processedData

Available Columns

category
day
dayOfWeek
latitude
longitude
month
pdDistrict
time
year

Query Search

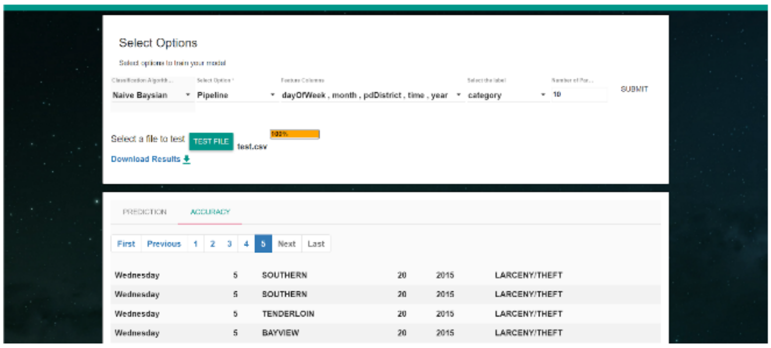
Enter a query to find values in the dataset using the table name given by you to preprocess dataset.

MINOR CRIMES	13	Wednesday	37.7547456	-122.3986956	5	BAYVIEW	19	2015
ASSAULT	13	Wednesday	37.7547456	-122.3986956	5	BAYVIEW	19	2015
LARCENY/THEFT	13	Wednesday	37.78730642	-122.4032848	5	SOUTHERN	19	2015
LARCENY/THEFT	13	Wednesday	37.77739182	-122.3976156	5	SOUTHERN	19	2015
SUSPICIOUS OCC	13	Wednesday	37.77176907	-122.4203472	5	SOUTHERN	19	2015
LARCENY/THEFT	13	Wednesday	37.78800653	-122.3998021	5	SOUTHERN	19	2015

Figure 4.16: Web application interface for querying data.

4.5.3 Prediction

Prediction provides the page to train the model using three different algorithms, namely Multilayer Perceptron Classification, Random Forest, and Nave Bayesian. After training the model user can upload a test file on which this model can be used to return a prediction. The results will be shown on the page itself and also an option to download the results as a text file is also provided. This is shown in Figure 4.17.



Select Options

Select options to train your model

Classification Algorith...: Naive Baysian
Select Option *: Pipeline
Feature Columns: dayOfWeek , month , pdDistrict , time , year
Select the label: category
Number of Par...: 10

Select a file to test test.csv

Download Results

Wednesday	5	SOUTHERN	20	2015	LARCENY/THEFT
Wednesday	5	SOUTHERN	20	2015	LARCENY/THEFT
Wednesday	5	TENDERLOIN	20	2015	LARCENY/THEFT
Wednesday	5	BAYVIEW	20	2015	LARCENY/THEFT

Figure 4.17: Web application interface for predicting category.

4.5.4 Prescription

4.5.4.1 District Boundaries

This page offers the user with the option of providing the required number of police districts required and give the population of the area and then using the redwing algorithm this page provides the user with police boundaries which are compact. A legend will provide the ID of different police districts according to the color as shown in Figure 4.18. Evaluation option provides the user with the option of seeing the population distribution within each district as in Figure 4.19.

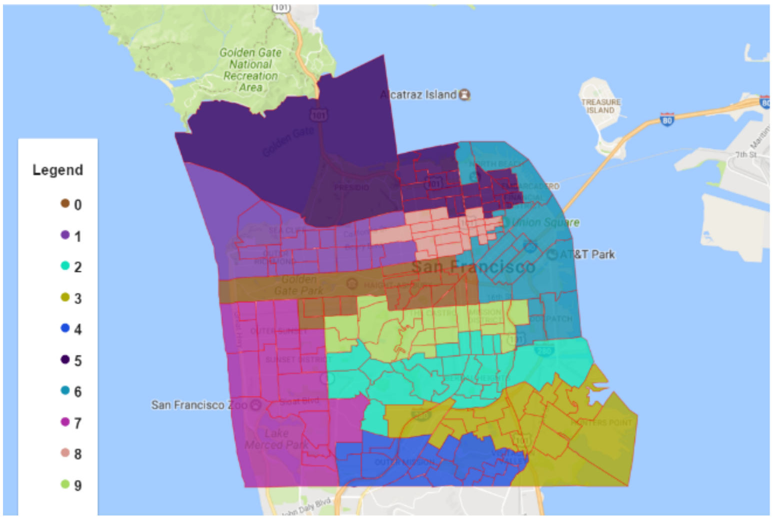


Figure 4.18: Redefining police district boundaries interface.

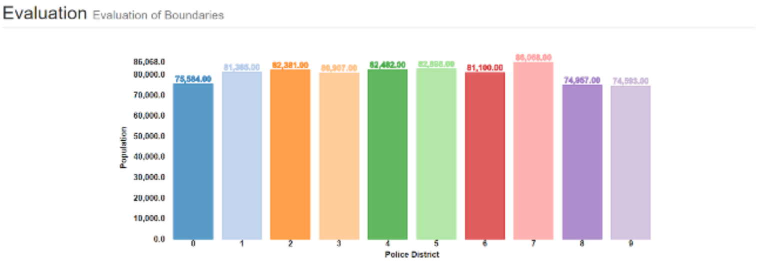


Figure 4.19: Evaluation of police district boundaries interface

4.5.4.2 Redefining Police Patrol Beats

It will take the user to a new page which would provide the user the option of generating new and efficient police beats for a given district ID from the police districts that were generated earlier. This is shown in Figure 4.20. Evaluate Results page provide three graphs showing the response time of each police beat, workload distribution among the beats and also compactness of each of beats.

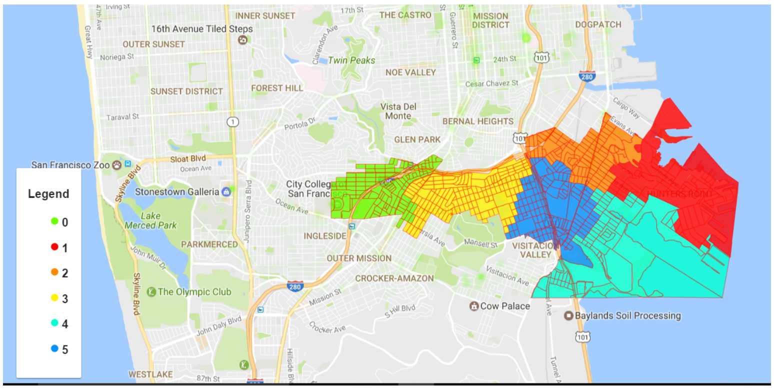


Figure 4.20: Redefining police patrol beats interface.

5. Summary

5.1 Conclusion

Most patrol beats used around the world today were drawn tens of years ago. Traditionally, geographic patrol boundaries and police district boundaries are drawn by hand based on a police departments knowledge, experience, and the available police resources [43]. This approach is error prone, because with the time possible crime locations can change due to the changes of physical structures, variation of populations in the area, and socioeconomic factors. Considerable variation can be seen even within a day (daytime vs. night) or a week (weekdays vs. weekends)

We have proposed a heuristic based, clustering method to divide a given police district into optimal patrol beats based on crime and census data. We use past crime data, their severity, census data, and geographic data to determine patrol beats and optimum seed points (i.e., optimum positioning of the police patrol car).once the police district boundaries are redefined, using our platform the relevant parties can rationalize police patrol beats inside each police district. Our system explore a way to find optimum patrol beat boundaries for a given police jurisdiction area based on the crimes happened earlier and the severity of them. Geometry is used to partition patrol beats appropriately. The police patrol beats are generated in a way so that they carry an equitable workload and minimize response time. To measure the inequality of workload distribution of newly created police patrol beats we have used Gini coefficient. The calculated Gini coefficient for new workload distribution in police patrol beats for 10 police districts was 0.036 and it suggests that the workload variance among districts is low. Also, as the compactness measure we have used Isoperimetric

Quotient and we got an average value of 0.71 for each police patrol beat and it suggests that the generated police patrol beats are more compact, and therefore it can minimize the police response time as expected. It can be proved by observing the average time taken for a police patrol car to arrive at any point inside the police patrol beats which is 25.22 seconds.

Apart from that we implemented a new heuristic-based spatial clustering approach to propose new police district boundaries. Here each district is represented as a cluster. In this approach we have used a target function which consists of population, compactness, and cost of growth of the cluster. By that we were able to generate compact police districts with equally distributed population. As a case study, we have applied our algorithm to redefine police district boundaries of San Francisco County. To measure the inequality of population distribution of newly created police districts we have used Gini Coefficient. The calculated Gini Coefficient for new population distribution was 0.02468 and it suggests that the population variance among districts is low. Also, as the compactness measure we have used Isoperimetric Quotient and we got an average value of 0.4957 for each district and which suggests the generated districts are compact. Hence, we can conclude that our algorithm has succeeded in the process of redistricting police district boundaries by generating more compact and equally distributed by population.

We developed a platform for crime data analysis which provides facilities for descriptive, predictive, and prescriptive analytics. With the backend support provided by Apache Spark, our platform is capable of handling large volume of crime records and analyze them efficiently. Through the web interface we provide, user can easily interact with the system and do the analytical work on a given crime dataset. Also, user is provided with visualization of results generated by the platform. The modularity of our system provides the option of writing new components without affecting the whole system. The features provided by our platform can be easily extended through this facility according to the various user requirements.

Platform also provides features to visualize crime data to analyse it descriptively. Heatmaps, Histograms, and interfaces to type in queries to the system are implemented in the system. A user is able to observe how crimes would happen around special events like New Years eve. It also provides a feature to observe how crimes have increased with years. Platform visualizes which crime categories have been more active in each year or frequency of crime categories for a given year using histograms. Meanwhile predictive analysis also helps a user to predict which crime category would most probably occur for a given time and area.

5.2 Future Work

As the data size and the covering geographical area is increased the solution provided by the Crime Data Analytic Platform needs more computation power. So it needs various parallelized and distributed system techniques. System currently uses crime data, census block data, census tract data, population data and race data to do data mining, prediction, police district boundary generation and patrol beats generation. But system can be extended to integrate other relevant data like offender residence,serial killers data.

Also the proposed algorithm can be further improved to consider about the prioritization of calls for services. The proposed algorithm can also be improved to provide police patrol routes, rather than an optimal position for a police car to locate. This can be done by considering different time periods, seasons, and also special occasions like New Year Eve. Furthermore using a proper GIS plan, the GIS data can be integrated with the crime data set. In this way, we can significantly improve the presision and the recall of the prediction model trained by the CDAP.

When considering about the visualizer, the capability of it can be greatly improved by integrating more visualizing models such as various graph types. Also binding data from backend to front end can be optimized further and it will improve the user experience significantly, Also by providing interactive guidance to the user, it will make the platform to be used by anyone with or without domain knowledge.

References

[1] S.K.Lodha and A.K.Verma, "Spatio-temporal visualization of urban crimes on a gis grid," *8th ACM Intl. symposium on Advances in Geographic Information Systems*, pp. 174–179, 2000.

[2] J. Forgeat. (2015) Data processing architectures lambda and kappa. [Online]. Available: https://www.ericsson.com/research-blog/data-knowledge/data-processing-architectures-lambda-and-kappa/

[3] C. Yu, M. W. Ward, M. Morabito, and W. Ding, "Crime forecasting using data mining techniques," *11th IEEE Intl. Conf. on Data Mining Workshops*, pp. 779–786, 2011.

[4] A. T. Murray, I. McGuffog, J. S. Western, , and P. Mullins, "Exploratory spatial data analysis techniques for examining urban crime implications for evaluating treatment," *British Journal of criminology*, vol. 41, no. 2, pp. 309–329, 2001.

[5] R. Krishnamurthy and J. S. Kumar, "Survey of data mining techniques on crime data analysis," *International Journal of Data Mining Techniques and Applications*, vol. 1, no. 2, pp. 117–120, 2012.

[6] D. E. Brown, "The regional crime analysis program (recap): a framework for mining data to catch criminals," *IEEE Intl. Conf. on Systems, Man, and Cybernetics*, vol. 3, pp. 2848–2853, 1998.

[7] H. Chen, D. Zeng, H. Atabakhsh, W. Wyzga, and J. Schroeder, "Coplink: managing law enforcement data and knowledge," *Communications of the ACM*, vol. 46, no. 1, pp. 28–34, 2003.

[8] A. Verma, R. Ramyaa, S. Marru, Y. Fan, and R. Singh, “Rationalizing police patrol beats using voronoi tessellations,” *IEEE Intl. Conf. on Intelligence and Security Informatics (ISI)*, pp. 165–167, 2010.

[9] J. E. Eck and D. L. Weisburd, “Crime places in crime theory,” *Crime and place: Crime prevention studies*, vol. 4, 2015.

[10] H. Chen, W. Chung, J. J. Xu, G. Wang, Y. Qin, and M. Chau, “Crime data mining: a general framework and some examples,” *Computer*, vol. 37, no. 4, pp. 50–56, 2004.

[11] P. S. Mitchell, “Optimal selection of police patrol beats,” *Criminal Law and Criminology*, vol. 63, p. 577, 1972.

[12] A.S.Foundation. (2015) Apache storm. [Online]. Available: http://storm.apache.org/

[13] S.Gopalani and R.Arora, “Comparing apache spark and map reduce with performance analysis using k-means,” *International Journal of Computer Applications*, vol. 113, no. 1, pp. 8–11, 2015.

[14] X. M. et al., “Mllib: Machine learning in apache spark,” *Machine Learning Research*, vol. 17, no. 34, pp. 1–7, 2015.

[15] Spark.apache.org. (2016) Decision trees - spark.mllib - spark 1.6.2 documentation. [Online]. Available: https://spark.apache.org/docs/1.6.2/mllibdecision-tree.html

[16] R.Iqbal, M.A.A.Murad, A.Mustapha, P.H.S.Panahy, and N.Khanahmadliravi, “An experimental study of classification algorithms for crime prediction,” *Indian Journal of Science and Technology*, vol. 6, no. 3, pp. 4219–4225, 2013.

[17] Saedsayad.com. (2016) Decision trees. [Online]. Available: http://www.saedsayad.com/decisiontree.htm

[18] Stat.berkeley.edu. (2016) Random forests - classification description. [Online]. Available: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

[19] Hiit.fi. (2016) Multilayer perceptrons. [Online]. Available: https://www.hiit.fi/u/ahonkela/dippa/node41.html

[20] Spark.apache.org. (2016) Frequent pattern mining - rdd-based api - spark 2.0.0 documentation. [Online]. Available: https://spark.apache.org/docs/latest/mllibfrequent-pattern-mining.htm

[21] J. P. J. Han and Y. Yin, "Mining frequent patterns without candidate generation," *ACM SIGMOD Record*, vol. 29, no. 2, pp. 1–12, 2000.

[22] M.Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss," *Machine learning*, vol. 29, no. 2–3, pp. 103–130, 1997.

[23] R. R. A. Verma and S. Marru, "Validating distance decay through agent based modeling," *Security Informatics*, vol. 2, no. 1, p. 1, 2013.

[24] R. Boba, "Introductory guide to crime analysis and mapping," *Community Oriented Policing Services.*, 2001.

[25] X. Wu, "An informative and predictive analysis of the san francisco police department crime data," 2016.

[26] S. I. Gass, "On the division of police districts into patrol beats," *23rd ACM National Conference*, pp. 459–473, 1968.

[27] S. J. DAmico, S. Wang, R. Batta, and C. M. Rump, "A simulated annealing approach to police district design," *Computers and Operations Research*, vol. 29, no. 6, pp. 667–684, may 2002.

[28] C. D. Hale, *Police Patrol, Operations, and Management.* Upper Saddle River, New Jersey: Prentice-Hall, Inc, 1980.

[29] F. Q. Curtin, K. Hayslett-McCall, "Determining optimal police patrol areas with maximal covering and backup covering location models," *Networks and Spatial Economics*, vol. 10, pp. 125–145, 2010.

[30] D. Joshi, L. Soh, , and A. Samal, "Redistricting using constrained polygonal clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 11, pp. 2065–2079, Nov 2012.

[31] E. Weisstein. Isoperimetric quotient. [Online]. Available: http://mathworld.wolfram.com/IsoperimetricQuotient.html

[32] E. W. Weisstein. (2000) Gini coefficient. [Online]. Available: http://mathworld.wolfram.com/GiniCoefficient.html

[33] www.graphhopper.com. Graphhopper routing engine. [Online]. Available: https://github.com/graphhopper/graphhopper

[34] (2015) Sf open data. [Online]. Available: https://data.sfgov.org/

[35] E. Weisstein. Gini coefficient. [Online]. Available: http://mathworld.wolfram.com/GiniCoefficient.html

[36] Y. Zhang, S. H. Huddleston, D. E. Brown, and G. P. Learmonth, "A comparison of evaluation methods for police patrol district designs," in *Proc. 2013 Winter Simulation Conference (WSC)*, 2013, pp. 2532–2543.

[37] (2015) Us census burearu. [Online]. Available: https://www.census.gov/

[38] Spark.apache.org. (2016) Feature extraction and transformation - rdd-based api. [Online]. Available: http://spark.apache.org/docs/latest/mllib-feature-extraction.html

[39] Spark ml library guide. [Online]. Available: http://spark.apache.org/docs/latest/ml-guide.html

[40] A.S.Bharathi and R.Shilpa, “A survey on crime data analysis of data mining using clustering techniques,” *International Journal of Advance Research in Computer Science and Management Studies*, vol. 2, no. 8, 2014.

[41] Spark.apache.org. (2016) Ensembles - mllib - spark 1.5.1 documentation. [Online]. Available: https://spark.apache.org/docs/1.5.1/mllib-ensembles.html

[42] Stat.berkeley.edu. (2016) Random forests - classification description. [Online]. Available: https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

[43] P. S. Mitchell, “Optimal selection of police patrol beats,” *J. Crim. L. Criminology, and Police Sci.*, vol. 63, p. 577, 1972.