**Towards fulfillment for Undergraduate Degree level Programmed Bachelor of Technology in Computer Engineering**

A Project Report on:

**“Enhancing Data Analytics in Smart City Infrastructure”**

Prepared By:

**Admission No Student Name**

**U13CO005 Heet Sheth**

**U13CO006 Rohit Maurya**

**U13CO007 Arshpreet Singh Pabla**

**U13CO013 Harshal Rohit**

Class : B.TECH. IV (Computer Engineering) 7th Semester

Year : 2016-2017

Guided By : **Dr. Dhiren R. Patel**



DEPARTMENT OF COMPUTER ENGINEERING

SARDAR VALLABHBHAI NATIONAL INSTITUTE OF TECHNOLOGY, SURAT – 395 007 (GUJARAT, INDIA)

Student Declaration

This is to certify that the work described in this project report has been actually carried out and implemented by our project team consisting of

|  |  |  |
| --- | --- | --- |
| **Sr.** | **Admission No.** | **Student Name** |
| 1 | U13CO005 | Heet Sheth |
| 2 | U13CO006 | Rohit Maurya |
| 3 | U13CO007 | Arshpreet Singh Pabla |
| 4 | U13CO013 | Harshal Rohit |

Neither the source code there in, nor the content of the project report have been copied or downloaded from any other source. We understand that our result grades would be revoked if later it is found to be so.

**Signature of the Students:**

|  |  |  |
| --- | --- | --- |
| **Sr.** | **Student Name** | **Signature of the Student** |
| 1 | Heet Sheth |  |
| 2 | Rohit Maurya |  |
| 3 | Arshpreet Singh Pabla |  |
| 4 | Harshal Rohit |  |

***Certificate***

This is to certify that the project report entitled “Enhancing Data Analytics in Smart City Infrastructure” is prepared and presented by

|  |  |  |
| --- | --- | --- |
| Sr. | Admission No. | Student Name |
| 1 | U13CO005 | Heet Sheth |
| 2 | U13CO006 | Rohit Maurya |
| 3 | U13CO007 | Arshpreet Singh Pabla |
| 4 | U13CO013 | Harshal Rohit |

Final Year of Computer Engineering and their work is satisfactory.

SIGNATURE:

GUIDE JURY HEAD OF DEPT.

Abstract

A large amount of land-use, environment, socio-economic, energy and transport data is generated in cities. An integrated perspective of managing and analyzing such big data can answer a number of science, policy, planning, governance and business questions and support decision making in enabling a smarter environment. This report presents a theoretical and experimental approach to enhance the data analytics layer in the smart city infrastructure framework. Currently, cursory inferences are being made on the available data which are only used for visualization. In addition to this, the proposed scheme would help to improve the quality of the data and extract some hidden, useful insights which will contribute in building a smarter city.

**Keywords:** Smart city - Analytics - Big Data - Machine Learning - Layered Infrastructure – Elasticsearch - Matlab

Table of Contents

[Table of Figures V](#_Toc482033493)

[List of Acronyms VI](#_Toc482033494)

[Chapter 1: Introduction 1](#_Toc482033495)

[1.1 Applications 1](#_Toc482033496)

[1.2 Motivation 2](#_Toc482033497)

[1.3 Objectives 3](#_Toc482033498)

[Chapter 2: Smart City 4](#_Toc482033499)

[2.1 Smart Cities Mission 4](#_Toc482033500)

[2.2 Features of a Smart City 4](#_Toc482033501)

[2.3 Smart City Infrastructure Components 5](#_Toc482033502)

[2.4 Surat as a smart city 7](#_Toc482033503)

[2.7 Existing Infrastructure 8](#_Toc482033504)

[Chapter 3: Proposed Layered Framework 10](#_Toc482033505)

[Layer 1: Information from various sources 10](#_Toc482033506)

[Layer 2: Middleware i.e. Platform Neutral API application 11](#_Toc482033507)

[Layer 3: Domain Related Applications 11](#_Toc482033508)

[Layer 4: Visualization and Analytics 11](#_Toc482033509)

[Chapter 4: Visualization (ELK Stack) 15](#_Toc482033510)

[4.1 Motivation: 15](#_Toc482033511)

[4.2 Why Elasticsearch? 16](#_Toc482033512)

[4.3 Inside the Elasticsearch 17](#_Toc482033513)

[4.3.1 Cluster and Nodes 18](#_Toc482033514)

[4.3.2 Index and Shards 19](#_Toc482033515)

[4.3.3 Data as a Document 20](#_Toc482033516)

[4.4 Document Metadata 21](#_Toc482033517)

[4.5 Distributed Data Storage in Elasticsearch 22](#_Toc482033518)

[4.6 Search in Elasticsearch 23](#_Toc482033519)

[4.6.1 The Empty Search 23](#_Toc482033520)

[4.6.2 Multi-index, Multi-type search 23](#_Toc482033521)

[4.6.3 Query DSL search 24](#_Toc482033522)

[4.7 Kibana 25](#_Toc482033523)

[4.8 Logstash 25](#_Toc482033524)

[Chapter 5: Data Analytics 26](#_Toc482033525)

[5.1 Defining the use-cases 26](#_Toc482033526)

[5.1.1 Waste Management (Garbage Collection): 27](#_Toc482033527)

[5.1.2 Monitoring gender equality: 28](#_Toc482033528)

[5.1.3 Water Management (Water Supply Requirement): 28](#_Toc482033529)

[5.1.4 Revenue Collection (Property Tax collection): 29](#_Toc482033530)

[5.2 An architecture of Data Analytics 30](#_Toc482033531)

[5.3 Data Collection 30](#_Toc482033532)

[5.4 Data Cleaning 33](#_Toc482033533)

[5.5 Feature Engineering 33](#_Toc482033534)

[5.5.1 Garbage System 33](#_Toc482033535)

[5.5.2 Gender Equality 35](#_Toc482033536)

[5.5.3 Water Supply Management 37](#_Toc482033537)

[5.5.4 Property Tax Collection 37](#_Toc482033538)

[5.6 Data Preprocessing 38](#_Toc482033539)

[5.6 Analytics on Data 40](#_Toc482033540)

[Chapter 6: Practical Implementation 43](#_Toc482033541)

[Chapter 7: Conclusion 46](#_Toc482033542)

[Chapter 8: Future Work 47](#_Toc482033543)

[References 48](#_Toc482033544)

[Acknowledgement 50](#_Toc482033545)

# Table of Figures

[Figure 1: Smart physical Infrastructure 6](#_Toc482038494)

[Figure 2: Introduced Layered Architecture 10](file:///E:\8th%20sem\project\report_v4.docx#_Toc482038495)

[Figure 3: Layer 4 aka Analytics cycle 12](#_Toc482038496)

[Figure 4: Visualization Methods 12](#_Toc482038497)

[Figure 5: Outlier Detection with real-world instances 13](#_Toc482038498)

[Figure 6: Benefits of elasticsearch 16](#_Toc482038499)

[Figure 7: Representation of cluster & nodes 17](#_Toc482038500)

[Figure 8: Representation of Primary and Replica shards inside a node 19](file:///E:\8th%20sem\project\report_v4.docx#_Toc482038501)

[Figure 9: Architecture of Data Analytics 29](#_Toc482038502)

[Figure 10: An example of time series data [21] 30](#_Toc482038503)

[Figure 11: Sources of data attributes [22] [23] [24] [25] 31](#_Toc482038504)

[Figure 12: Major Data Sources 31](#_Toc482038505)

[Figure 13: Feature Vector - Garbage collection 33](#_Toc482038506)

[Figure 14: Surat divided into 7 zones 34](#_Toc482038507)

[Figure 15: Feature Vector - Gender Equality 34](#_Toc482038508)

[Figure 16: Gender Equality Ratio of Surat and each Zones 35](#_Toc482038509)

[Figure 17: Feature Vector - Water Supply 36](#_Toc482038510)

[Figure 18: Feature Vector – Property Tax Collection 36](#_Toc482038511)

[Figure 19: Data Preprocessing - Standardization of zone codes 38](#_Toc482038512)

[Figure 20: Data Preprocessing - Separation of date from date-stamp 38](#_Toc482038513)

[Figure 21: Data Preprocessing - Representation of Months 39](#_Toc482038514)

[Figure 22: Random data points and linear regression 40](#_Toc482038515)

[Figure 23: Support Vector Machine 41](#_Toc482038516)

[Figure 24: Developed dashboard for a polutants data 42](#_Toc482038517)

[Figure 25: Performance table - Garbage Collection - No. of containers 43](#_Toc482038518)

[Figure 26: Performance table - Garbage Collection - Weight collected (in Mt.) 43](#_Toc482038519)

[Figure 27: Data Analytics - Gender Equality Performance (City and all zones) 44](#_Toc482038520)

[Figure 28: Data Analytics - Property Tax (Advance Payment) 45](#_Toc482038521)

[Figure 29: Data Analytics - Property Tax (On-time payment) 45](#_Toc482038522)

[Figure 30: Data Analytics - Property Tax (Late payment) 45](#_Toc482038523)

# List of Acronyms

1. ICT: Information and Communication Technology
2. IoT: Internet of Things
3. ULB: Urban Local Body
4. KPI: Key Performance Indicator
5. MIT: Massachusetts Institute of Technology
6. DCS: District Cooling System
7. GIFT city: Gujarat International Finance Tec-City
8. SMC: Surat Municipal Corporation
9. SCP: Smart City Proposal
10. WSN: Wireless Sensor Networks
11. GUI: Graphical User Interface
12. API: Application Program Interface
13. ELK: Elasticsearch Logstash Kibana

# Chapter 1: Introduction

There has been a lot of activity around the concept of Smart City for some time. Cities are being identified as future smart cities. Theoretically at least, smart cities can fundamentally change our lives at many levels such as less pollution, garbage disposal, parking problems and more energy savings. Though the prospect seems mouth-watering, the implementation of the smart city concept around the world has been sporadic at best because of several reasons. Smart cities faced serious challenges prior to widespread acceptance, but their integrated use of Data analytics, IoT, and other technologies to solve contemporary urban issues eventually lead to their adoption. Whatever the stage the smart city implementation is at globally, data analytics and the IoT have the power to drive the implementation and are going to work with other software and hardware to lead the vision of smart city to fruition.

## 1.1 Applications

Some of the best ways in which analytics could have a profound effect on ‘Smart Cities’ are:

1. **Security-** Predictive Analytics have been used in several cities across the world to help predict where crimes are likely to take place through historical and geographical data. Through data, it is often not even necessary to make arrests, having police officers appearing in certain areas at specific times have seen crime rates drop.
2. **City Planning:** It allows models to be built to maximize the accessibility of certain areas or services whilst minimizing the risk of overloading important elements of the city’s infrastructure. In Short, it creates efficiency. Using data and modeling it is possible to map the infrastructure outcomes of any use of space with a high degree of accuracy.
3. **Transport:** Through using data throughout a transport network, we can create effective and flexible public transport, decreasing delays and increasing efficiency. Using data to not only predict when peak times will be for upcoming events, but to help monitor equipment will mean that reliability will improve and accidents will decrease.
4. **Future Proofing:** Through the use of modeling and predictive analytics, it becomes possible for city planners to see where the areas of growth are likely to be and how large the increase will be. Amenities can then be upgraded to accommodate this. In this way growth in certain areas can continue without the need for services to catch up.
5. **Web Provision:** The ability to shift bandwidth within a city in order to provide fast web access in the correct areas and for the correct people can play a key role in smart city development. For instance, if an area wants to attract more high-tech industries and web development companies, allowing bandwidth to be higher in those areas is going to be important and data modeling will allow this to be done most effectively.
6. **Sustainability:** One of the keys to sustainability is monitoring and having effective controls in place to quickly make changes in order to keep output at a certain level. Data is the most decisive factor here, it allows for governments and companies to see how their outputs are having a positive or negative result on the city as a whole. Thus, data monitoring and processing creates the opportunity to see which technologies work best and what new innovations could be used in particular areas in order to prevent any potential damage.
7. **Effective Spending:** Using analytics and data, ‘Smart Cities’ can target where the public money would have the most impact and what work would be most adequate for that. Through targeting where money would be best spent, the entire infrastructure of the city can be improved and wastage of money can be minimized.

## 1.2 Motivation

New data is being generated on an enormous scale and nowhere more so than in cities. Many everyday activities and interactions within urban environments can be harnessed to generate a variety of data, ranging from apps and social media to satellite imagery. It is not just the quantity but also the veracity of data which is increasing – emergent information technologies show what people do, not just what they say they do. However, to make sense of this vast data-flow, it is necessary to develop better ways to analyze it. Just as important then as new data-generating technologies are emerging new methods of analytics, such as machine learning, data mining, pattern recognition, profiling, and simulation and optimization algorithms. Cities can increasingly be seen as a key challenge – a modern-day ‘enigma code’. Harnessing data and analytics to make sense of the patterns within this code and thus cracking the ‘science of cities’ is a key future challenge.

## 1.3 Objectives

The framework of the Smart Cities thematic strand shares a holistic view over the city’s systems where efficiency, optimization, and data integration are common practices, and the understated objectives are met:

* develop solutions to attain efficient, resilient and optimized ICT urban network
* ascertain how to provide and use data generated by ICT to increase the understanding of the urban space and how it is lived by people
* research and develop an architecture for the IoT in sustainable cities
* implement suitable algorithms, machine learning and pattern recognition techniques for urban human motion analysis and visualization
* interpret, visualize the techniques and practices to enable human analysis of multi-dimensional urban data sets
* analyze and understand the human interaction with urban space, by considering the ergonomics of the urban structures

# Chapter 2: Smart City

A smart city is an urban development vision to integrate multiple ICT and IoT solutions in a [secure](https://en.wikipedia.org/wiki/Information_security) fashion to manage a city’s assets – the city’s assets include, but are not limited to, local departments' information systems, schools, libraries, transportation systems, hospitals, power plants, water supply networks, waste management, law enforcement, and other [community services](https://en.wikipedia.org/wiki/Community_service). [4]

## 2.1 Smart Cities Mission

Smart Cities Mission is an urban renewal and retrofitting program by the Government of India with a mission to develop 100 cities (the target has been revised to 109 cities) all over the country making them citizen friendly and sustainable. The Union Ministry of Urban Development is responsible for implementing the mission in collaboration with the state governments of the respective cities. The government of India under Prime Minister Narendra Modi has a vision of developing 100 smart cities as satellite towns of larger cities and by modernizing the existing mid-sized cities.

The goal of building a smart city is to improve [quality of life](https://en.wikipedia.org/wiki/Quality_of_life) by using urban informatics and technology to improve the efficiency of services and meet residents’ needs. ICT allows city officials to interact directly with the community and the city infrastructure and to monitor what is happening in the city, how the city is evolving, and how to enable a better quality of life. Through the use of sensors integrated with real-time monitoring systems, [data are collected](https://en.wikipedia.org/wiki/Data_collection) from citizens and devices - then processed and analyzed. The information and knowledge gathered are keys to tackling inefficiency.

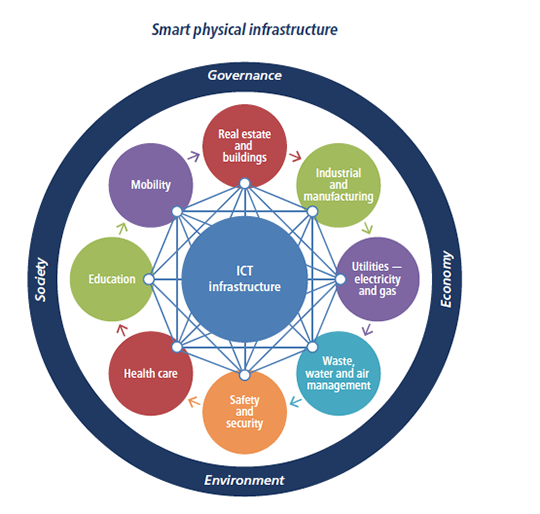
## 2.2 Features of a Smart City

There are a variety of smart city features such as Promoting mixed land use in area based developments, Housing and inclusiveness, Creating walkable localities, Preserving and developing open spaces etc. However, we will be primarily focusing on

* Making governance citizen-friendly and cost effective- increasingly rely on online services to bring about accountability and transparency, especially using mobiles to reduce cost of services and providing services without having to go to municipal offices. Forming e-groups to listen to people and obtain feedback and use online monitoring of programs and activities with the aid of cyber tour of worksites. Modern digital infrastructure, combined with a secure but open access approach to public re-useable data, which enables citizens to access the information they need, when they need it.
* A recognition that service delivery is improved by being citizen centric: this involves placing the citizen’s needs at the forefront, sharing management information to provide a coherent service, rather than operating in a multiplicity of service silos (for example, sharing changes of address more effectively), and offering internet service delivery where possible (at a fraction of the face to face cost).
* An intelligent physical infrastructure (“smart” systems or the Internet of Things), to enable service providers to use the full range of data both to manage service delivery on a daily basis and to inform strategic investment in the city/community. (for example, gathering and analyzing data on whether public transport is adequate to cope with rush hour peaks)

## 2.3 Smart City Infrastructure Components

Infrastructure is the foundation for the development of a smart city. Smart Infrastructure can be broadly divided into two categories: (1) physical and (2) digital. Smart physical infrastructures include: (1) Smart Buildings, (2) Smart Mobility and Transport, (3) Smart Energy, (4) Smart Water Management, (5) Smart Waste Management and (6) Smart Healthcare as explained in the figure [1]. In terms of the digital infrastructure, ICT and Data infrastructure enable a digital platform from which an information and knowledge network can be created. This facilitates the aggregation of city information for data analysis and also can be used to better understand how the city is functioning. One way to look at the digital infrastructure is in the form of different digital supporting layers such as (1) Urban layer, (2) Sensor layer, (3) Connectivity layer, (4) Data analytics layer and (5) Automation layer is also presented with case studies and examples. The chapter concludes by highlighting the need for an integrated approach in dealing with these diverse smart city infrastructure components.



**Figure 1: Smart physical Infrastructure**

#### The Need for an Integrated Approach for Smart Infrastructure

A city has many physical functions which are manifested in the different forms of infrastructure – water, waste, buildings and so on. Each infrastructure element is a system and is made up of subsystems, components and devices which behave like a communications data network by communicating between themselves. The city is made up of these different infrastructure verticals forming a “system of systems”. There is a clear nexus between these different systems, for example a building uses energy, water and generates waste and if the different individual smart systems come together, the building becomes “smart”. However, in many cases these city infrastructure elements typically tend to operate in silos. Smart Cities needs an integrated treatment of all smart infrastructures. Smarter ways to develop cities will emerge when city governments and citizens start thinking and planning for infrastructure components in a holistic manner. One such example of an integrated approach is explained below:

#### EXAMPLE:

**Co-locating smart infrastructure, GIFT City, India**

The Gujarat International Finance Tec-City (GIFT City) is a good example of an integrated smart city – incorporating the different functions of a city – Water, Waste, Air-Conditioning, Energy / Utilities, Transport and an ICT backbone to help manage these components. It has built an infrastructure sufficient for 62 million square feet of real estate on an 886 acre land bank. Power and other utilities including the chilled water for the district cooling system (DCS) are all placed all in a common tunnel accessible to material handling and maintenance vehicles. The tunnel has a cross section of 7.6-by-6.2 meters and is 15 km in length. The tunnel will carry 50 million liters per day of treated water, 200,000 tons of cooling water, 750 megawatts of power supply, and 280 million tons of solid waste. This integrated tunnel approach saved 1,500 man-hours and INR 900 million in design costs alone. Such a holistic and integrated approach of infrastructure applications is creating a new paradigm and benchmark in city development.

### 

### 2.4 Surat as a smart city

Smart Cities Mission was launched by Prime Minister Shri Narendra Modi on 25 June, 2015. Surat city was selected among 100 cities to be developed as smart city in India due to various achievements, initiatives and all inclusive approaches. Accordingly, Surat city had submitted “Smart City Proposal” (SCP) for Surat City in the given format on 15 December, 2015 to Ministry of Urban Development, Government of India with required consent of Government of Gujarat and statutory authority of Surat Municipal Corporation (SMC).

Citizens Engagement is the base of four pillars of institutional, physical, social and economic infrastructure for comprehensive development as per Smart City Mission guidelines and therefore Citizens Engagement tool is extensively used to know the suggestions/feedback from citizens by various online & offline methods like – Stakeholders consultation meetings with Elected representatives, Press media, different industrial, trade & commerce associations, doctors, engineers, architects and NGOs, Ward level meetings with citizens, Essay & Drawing competition for students and citizens, Techno fair for informing citizens about possible smart solutions, Citizens Poll on myGov and SMC’s website and seminars / webinars on different subjects etc.

Based on Citizens Poll for Pan City initiatives, ICT based Transport-Connectivity smart solutions are finalized in Smart City Proposal, which will be implemented in entire area of Surat city. Some instances of Smart City implementations in Surat are:

* Vehicle actuated system: operates by adapting to the volume of vehicles on the road to alleviate congestion, while still continuing to provide safety for pedestrians and bicyclists.
* Countdown timers on signals: Countdown timers at red lights warning drivers how much time is left before the light changes can help to ease traffic flow, residents and road safety experts say.
* S-Connect: Smart Card lets you pay the fine through a cashless transaction.
* Smart Street lights: street lamps light up automatically once it gets dark in the evening
* Logistic parks: it is a special economic zone with conducive work environment and other innovations to instill confidence among the business houses for a hassle-free trade

## 2.7 Existing Infrastructure

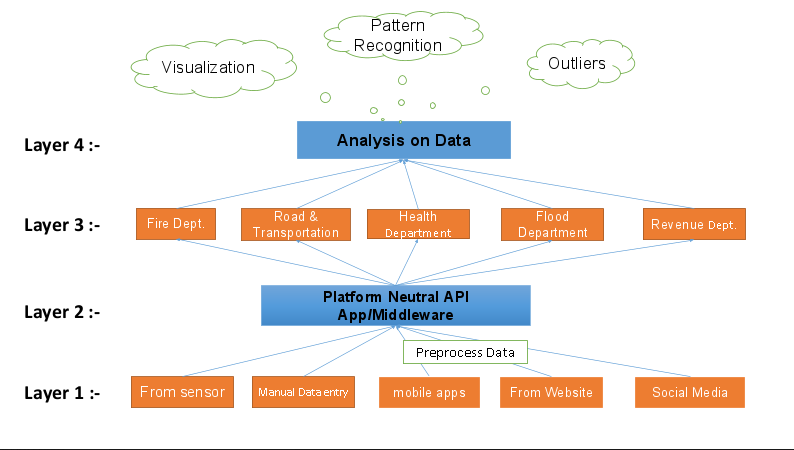
Currently SMC has developed many mobile applications and web interface to collect data from citizens as well as provide services in a much better and reliable way. Each application interacts with various departments such as water department, fire department etc. by providing relevant data to them. Each department requires its own set of data, and hence various APIs have been created to serve that purpose. Every department co-ordinates and manipulates its own database which is maintained by their respective administrators. Data format is also specific to the application in use.

#### Issues in this infrastructure:

The major issue in the current architecture is that there is no centralized layer to manage all the departments using a single interface. Managing and updating these APIs is very time consuming and adding a new API takes much more effort. Also, there is a lack of a standard data format as every department uses their own format. SMC is collecting lots of data which can be useful to make inferences by developing structured training models. However, at present no significant work is being done. The collected data is used only for visualization.

Due to lack of centralization, maintaining the current architecture requires much more cost and human effort.

# Chapter 3: Proposed Layered Framework

****To solve the issues currently faced by previously mentioned architecture, we have come up with a 4-layer framework that redefines the approach to build a Smart City.

**Figure 2: Introduced Layered Architecture**

## Layer 1: Information from various sources

A smart city can create an efficient and smart services delivery platform for public and municipal workers by installing sensors in the city and to create platforms that allow the share of information and give it for proper use to the public, city managers, businesses and professionals.

To harness the true potential of Smart cities, the city must become a platform i.e. an enabler for developers, creativity and applications. In doing so, the city becomes like the Internet i.e. a connector and an enabler for citizens which aims to empower the citizen.

Smart city applications are similar to conventional mobile applications. However, in conjunction with mobile devices which often include sensors, Smart city applications would interface with hardware – typically open source hardware – which enables ‘intelligent data’ spots in the city

The data from all the sources is sent to a single data repository in a predefined format as determined by Layer 2.

## Layer 2: Middleware i.e. Platform Neutral API application

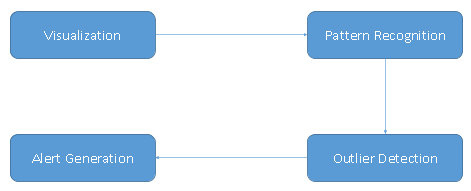
This layer consists of platform-neutral APIs which form a Remote control network which is an integrated control network with common data transmission infrastructure that monitors all the municipal and supply networks of the service companies involved in the project. The goal is to manage and route data related to a specific domain as required by the user. This layer is a GUI based platform.

## Layer 3: Domain Related Applications

Through the use of the APIs of layer 2, independent applications can extract the data related to their domain as and when required. Scope of APIs can easily be modified so that the application can adapt to changing requirements. Different departments can execute multiple instances of a single API concurrently.

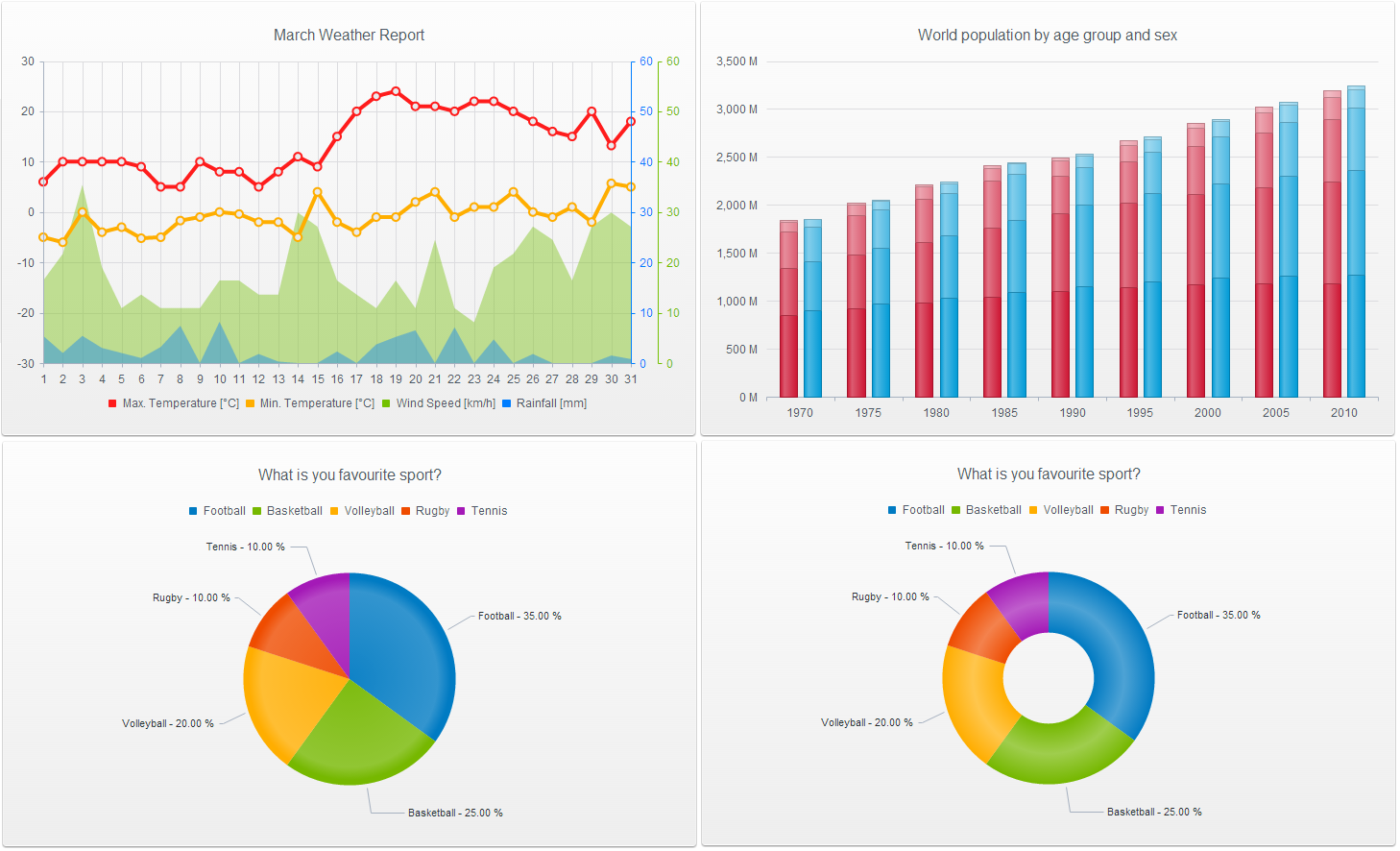
## Layer 4: Visualization and Analytics

In order to gather urban information, smart cities use elements of the Internet of Things (IoT), such as mobile phones, RFID cards and wireless sensor networks (WSNs). The data collected by these devices is used in a plethora of applications. For example, traffic monitoring sensors are used to control traffic lights and wireless meters are installed in pipes to monitor leaks and ruptures. Moreover, this data gives city managers and other stakeholders the opportunity to plan future facilities using a better picture of citizens’ behavior and the real use of the current infrastructures. However, to make use of this data we should be able to process it and perform the following operations.

****

**Figure 3: Layer 4 aka Analytics cycle**

**Visualization:** Data visualization is the presentation of data in a pictorial or graphical format. It enables decision makers to see analytics presented visually, so they can grasp difficult concepts or identify new patterns. With interactive visualization, you can take the concept a step further by using technology to drill down into charts and graphs for more detail, interactively changing what data you see and how it’s processed. Because of the way the human brain processes information, using charts or graphs to visualize large amounts of complex data is easier than poring over spreadsheets or reports. Data visualization is a quick, easy way to convey concepts in a universal manner–and you can experiment with different scenarios by making slight adjustments.



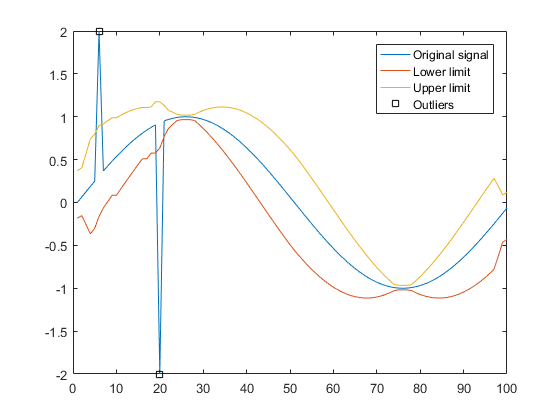
**Figure 4: Visualization Methods**

**Pattern Recognition:** The growth in the amount of available data makes necessary the development of tools to handle such huge data bases that, in addition, can be complex (time-series data or images). It is necessary to fully exploit this data by extracting all the useful information from it. However, it is impractical to manually analyze, explore, and understand the data. As a result, useful information is often overlooked, and the potential benefits of increased computational and data gathering capabilities are only partially attained.

Pattern recognition and data mining focus on the analysis of great amount of data. Pattern recognition comes from the engineering field and, although originally motivated by the analysis of images, it has currently extended its scope to the discovery of patterns in any kind of huge databases.

It is a branch of machine learning that focuses on the recognition of patterns and regularities in data. Stages in pattern recognition may involve measurement of the object to identify distinguishing attributes, extraction of features for the defining attributes, and comparison with known patterns to determine a match or mismatch.

**Outlier Detection:** This accelerated deployment of smart city technology has often resulted in leaving security aside as a secondary issue. In order to mitigate the control and visibility problems affecting the data quality of the smart city data collection systems, we need to incorporate outlier detection algorithms. Outlier detection is the identification of items, events or observations which do not conform to an expected pattern or other items in a dataset.  These algorithms will help detect if some wrongful data is being pushed from the sources and work towards removing such data.

****

**Figure 5: Outlier Detection with real-world instances**

# Chapter 4: Visualization (ELK Stack)

Elasticsearch is an open-source search engine built on top ofApache Lucene, a full-text search-engine library.Lucene is arguably the most advanced, high-performance, and fully featured search engine library in existence today - both open source and proprietary. Elasticsearch is also written in Java and uses Lucene internally for all of its indexing and searching, but it aims to make full-text search easy by hiding the complexities of Lucene behind a simple, coherent, RESTful API. However, Elasticsearch is much more than just Lucene and much more than “just” full-text search.

It can also be described as follows:

* A distributed real-time document store where every field is indexed and searchable.
* A distributed search engine with real-time analytics.
* Capable of scaling to hundreds of servers and petabytes of structured and unstructured data.

And it packages up all this functionality into a standalone server that your application can talk to via a simple RESTful API, using a web client from your favorite programming language, or even from the command line.

## Motivation:

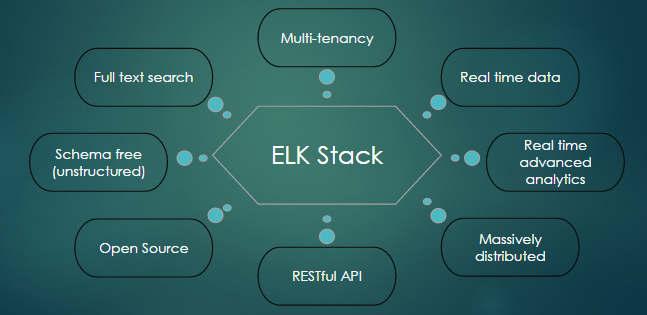
A huge amount of data is produced by various digital sources in Surat city which is being collected by Surat Smart City team. Here we are talking about an information/data ocean; we call this ocean big data in the world of information technology. A significant part of this data is unstructured, scattered and insignificant when it is alone.

For this reason, some requirements of this data are at stake, such as recording, accessing, analyzing and processing the data. Like similar search engines, Elasticsearch is a tool developed for dealing with the problems of big data mentioned above. Currently it is getting notice and it is used for not only structured search but also for analysis on data. Hence this leads to the motivation behind exploring Elasticsearch and get to know its functioning in a much detail.

## Why Elasticsearch?

The only reason why we are using elasticsearch is the abundant properties that Elasticsearch possess which will help us enhancing the overall platform.

1. **Real-Time Data:** How long can you wait for insights on your fast-moving data? With Elasticsearch, all data is immediately made available for search and analytics.
2. **Real-Time Advanced Analytics:** Combining the speed of search with the power of analytics changes your relationship with your data. Interactively search, discover, and analyze to gain insights that improve your products or streamline your business.
3. **Massively Distributed:** Elasticsearch allows you to start small and scale horizontally as you grow. Simply add more nodes, and let the cluster automatically take advantage of the extra hardware. Petabytes of data? Thousands of nodes? No problem.
4. **High Availability:** Elasticsearch clusters are resilient — they will detect new or failed nodes, and reorganize and rebalance data automatically, to ensure that your data is safe and accessible.
5. **Multitenancy:** A cluster may contain multiple indices that can be queried independently or as a group. Index aliases allow filtered views of an index, and may be updated transparently to your application.
6. **Full-Text Search:** Elasticsearch builds distributed capabilities on top of Apache Lucene to provide the most powerful full- text search capabilities available. Powerful, developer-friendly query API supports multilingual search, geolocation, contextual did-you-mean suggestions, autocomplete, and result snippets.
7. **Document-Oriented:** Store complex real world entities in Elasticsearch as structured JSON documents. All fields are indexed by default, and all the indices can be used in a single query, to easily return complex results at breathtaking speed.
8. **Schema-Free:** Elasticsearch allows you to get started fast. Simply index a JSON document and it will automatically detect the data structure and types, create an index, and make your data searchable. You also have full control to customize how your data is indexed.
9. **Developer-Friendly, RESTful API:** Elasticsearch is API driven. Almost any action can be performed using a simple RESTful API using JSON over HTTP. Client libraries are available for many programming languages.



**Figure 6: Benefits of elasticsearch**

1. **Per-Operation Persistence:** Elasticsearch puts your data safety first. Document changes are recorded in transaction logs on multiple nodes in the cluster to minimize the chance of any data loss.
2. **Apache 2 Open Source License:** Elasticsearch can be downloaded, used, and modified free of charge. It is available under the Apache 2 license, one of the most flexible open source licenses available.
3. **Build on top of Apache Lucene™:** Apache Lucene is a high performance, full-featured Information Retrieval library, written in Java. Elasticsearch uses Lucene internally to build its state of the art distributed search and analytics capabilities.

## Inside the Elasticsearch

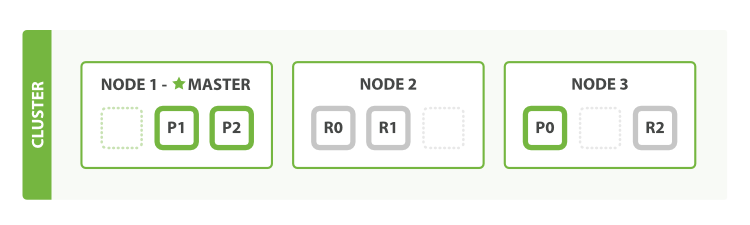
No individual part of Elasticsearch is new or revolutionary. Full-text search has been done before, as have analytics systems and distributed databases. The revolution is the combination of these individually useful parts into a single, coherent, real-time application.

Elasticsearch is built to be always available, and to scale with user’s needs. Scale can come from buying bigger servers (*vertical scale*, or *scaling up*) or from buying more servers (*horizontal scale*, or *scaling out*). While Elasticsearch can benefit from more-powerful hardware, vertical scale has its limits. Real scalability comes from horizontal scale—the ability to add more nodes to the cluster and to spread load and reliability between them.

With most databases, scaling horizontally usually requires a major overhaul of application to take advantage of these extra boxes. In contrast, Elasticsearch is distributed by nature: it knows how to manage multiple nodes to provide scale and high availability. This also means that our application doesn’t need to care about it.

In this section, the components such as cluster, nodes, and shards are explained and also how they scale with needs and to ensure that data is safe from hardware failure.

### Cluster and Nodes

A node is a running instance of Elasticsearch, while a cluster consists of one or more nodes with the same cluster.name that are working together to share their data and workload. As nodes are added to or removed from the cluster, the cluster reorganizes itself to spread the data evenly. One node in the cluster is elected to be the master node, which is in charge of managing cluster-wide changes like creating or deleting an index, or adding or removing a node from the cluster. The master node does not need to be involved in document-level changes or searches, which means that having just one master node will not become a bottleneck as traffic grows. Any node can become the master. Our example cluster which is shown in Figure-7 has three nodes.

**Figure 7: Representation of cluster & nodes**

As users, we can talk to any node in the cluster, including the master node. Every node knows where each document lives and can forward our request directly to the nodes that hold the data we are interested in. Whichever node we talk to manage the process of gathering the response from the node or nodes holding the data and returning the final response to the client. It is all managed transparently by Elasticsearch.

Many statistics can be monitored in an Elasticsearch cluster, but the single most important one is cluster health, which reports a status of green, yellow, or red. The status field provides an overall indication of how the cluster is functioning. The meanings of the three colours are provided here for reference:

**Green:** All primary and replica shards are active.

**Yellow:** All primary shards are active, but not all replica shards are active.

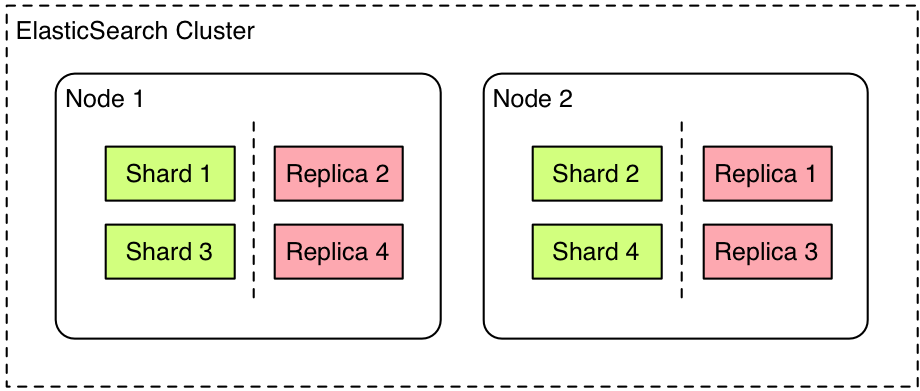
**Red:** Not all primary shards are active.

### Index and Shards

To add data to Elasticsearch, we need an index—a place to store related data. In reality, an index is just a logical namespace that points to one or more physical shards.

A shard is a low-level worker unit that holds just a slice of all the data in the index. Shard is a single instance of Lucene, and is a complete search engine in its own right. Our documents are stored and indexed in shards, but our applications don’t talk to them directly. Instead, they talk to an index.

Shards are how Elasticsearch distributes data around your cluster. Think of shards as containers for data. Documents are stored in shards, and shards are allocated to nodes in cluster. As cluster grows or shrinks, Elasticsearch will automatically migrate to shards between nodes so that the cluster remains balanced. Below figure gives an idea of primary and replica shards inside a cluster.

****

**Figure 8: Representation of Primary and Replica shards inside a node**

A shard can be either a primary shard or a replica shard. Each document in index belongs to a single primary shard, so the number of primary shards that you have determines the maximum amount of data that your index can hold.

A replica shard is just a copy of a primary shard. Replicas are used to provide redundant copies of data to protect against hardware failure, and to serve read requests like searching or retrieving a document.

The number of primary shards in an index is fixed at the time that an index is created, but the number of replica shards can be changed at any time.

### Data as a Document

Whatever program we write, the intention is the same: to organize data in a way that serves our purposes. But data doesn’t consist of just random bits and bytes. We build relationships between data elements in order to represent entities, or things that exist in the real world. A name and an email address have more meaning if we know that they belong to the same person.

In the real world, though, not all entities of the same type look the same. One person might have a home telephone number, while another person has only a cell-phone number, and another might have both. One person might have three email addresses, while another has none. A Spanish person will probably have two last names, while an English person will probably have only one. One of the reasons that object-oriented programming languages are so popular is that objects help us represent and manipulate real-world entities with potentially complex data structures.

The problem comes when we need to store these entities. Traditionally, we have stored our data in columns and rows in a relational database, the equivalent of using a spreadsheet. All the flexibility gained from using objects is lost because of the inflexibility of our storage medium. But what if we could store our objects as objects? Instead of modelling our application around the limitations of spreadsheets, we can instead focus on using the data. The flexibility of objects is returned to us.

An object is a language-specific, in-memory data structure. To send it across the network or store it, we need to be able to represent it in some standard format. JSON is a way of representing objects in human-readable text. It has become the de facto standard for exchanging data in the NoSQL world. When an object has been serialized into JSON, it is known as a JSON document.

Elasticsearch is a distributed document store. It can store and retrieve complex data structures—serialized as JSON documents—in real time. In other words, as soon as a document has been stored in Elasticsearch, it can be retrieved from any node in the cluster.

Of course, we don’t need to only store data; we must also query it. While NoSQL solutions exist that allow us to store objects as documents, they still require us to think about how we want to query our data, and which fields require an index in order to make data retrieval fast.

In Elasticsearch, all data in every field is indexed by default. That is, every field has a dedicated inverted index for fast retrieval. And, unlike most other databases, it can use all of those inverted indices in the same query, to return results at breathtaking speed.

## Document Metadata

A document doesn’t consist only of its data. It also has metadata—information about the document. The three required metadata elements are as follows:

\_index:- Where the document lives.

\_type:- The class of object that the document represents.

\_id:- The unique identifier for the document.

**\_index:** An index is a collection of documents that should be grouped together for a common reason. Elasticsearch indices, compared to database management systems, may be considered databases. As a database is a collection of regular information, Elasticsearch indices are collections of structured JSON documents. For example, you may store all your products in a products index, while all your sales transactions go in sales.

**\_type:** Data may be grouped loosely together in an index, but often there are sub-partitions inside that data which may be useful to explicitly define. Elasticsearch exposes a feature called types which allows you to logically partition data inside of an index. Documents in different types may have different fields, but it is best if they are highly similar. A \_type name can be lowercase or uppercase, but shouldn’t begin with an underscore or period. It also may not contain commas, and is limited to a length of 256 characters. Types can be considered tables, again compared to database management systems. Indices may contain one or more types. For example, all your products may go inside a single index. But you have different categories of products, such as "electronics", "kitchen" and "lawn-care".

**\_id:** The ID is a string that, when combined with the \_index and \_type, uniquely identifies a document in Elasticsearch. When creating a new document, you can either provide your own \_id or let Elasticsearch generate one for you.

**Mapping:** Mapping is the process of defining how a document should be mapped to the search engine. Types are created according to the mapping information. Elasticsearch creates mapping automatically (explicit mapping) based on the data sent (for example, string, integer, double, Boolean). You can override the default mapping by defining a new mapping.

## Distributed Data Storage in Elasticsearch

When you index a document, it is stored on a single primary shard. How does Elasticsearch know which shard a document belongs to? When we create a new document, how does it know whether it should store that document on shard 1 or shard 2?

The process can’t be random, since we may need to retrieve the document in the future. In fact, it is determined by a simple formula:

shard = hash(routing) % number\_of\_primary\_shards

The routing value is an arbitrary string, which defaults to the document’s \_id but can also be set to a custom value. This routing string is passed through a hashing function to generate a number, which is divided by the number of primary shards in the index to return the remainder. The remainder will always be in the range 0 to *number\_of\_primary\_shards-1*, and gives us the number of the shard where a particular document lives.

This explains why the number of primary shards can be set only when an index is created and never changed: if the number of primary shards ever changed in the future, all previous routing values would be invalid and documents would never be found.

## Search in Elasticsearch

Elasticsearch not only *stores* the document, but also *indexes* the content of the document in order to make it searchable. Every field in a document is indexed and can be queried. And it’s not just that. During a single query, Elasticsearch can use *all* of these indices, to return results at breath-taking speed. That’s something that you could never consider doing with a traditional database.

A search can be any of the following:

* A structured query on concrete fields like gender or age, sorted by a field like join\_date, similar to the type of query that you could construct in SQL.
* A full-text query, which finds all documents matching the search keywords, and returns them sorted by relevance.
* A combination of the two.

### The Empty Search

The most basic form of the search API is the *empty search*, which doesn’t specify any query but simply returns all documents in all indices in the cluster:

GET /\_search

### Multi-index, Multi-type search

Usually, however, you will want to search within one or more specific indices, and probably one or more specific types. We can do this by specifying the index and type in the URL, as follows:

/\_search: Search all types in all indices.

/ind1/\_search: Search all types in the ind1 index.

/ind1,ind2/\_search: Search all types in the ind1 and ind2 indices.

/g\*,u\*/\_search: Search all types in any indices beginning with g or beginning with u.

/ind1/user/\_search: Search type user in the ind1 index.

/ind1,ind2/user,tweet/\_search: Search types user and tweet in the ind1 and ind2 indices.

/\_all/user,tweet/\_search: Search types user and tweet in all indices

When you search within a single index, Elasticsearch forwards the search request to a primary or replica of every shard in that index, and then gathers the results from each shard. Searching within multiple indices works in exactly the same way—there are just more shards involved.

### Query DSL search

The query DSL (Domain Specific Language) is a flexible, expressive search language that Elasticsearch uses to expose most of the power of Lucene through a simple JSON interface. It is what you should be using to write your queries in production. It makes your queries more flexible, more precise, easier to read, and easier to debug.

To use the Query DSL, pass a query in the query parameter:

GET /\_search

{

"query": YOUR\_QUERY\_HERE

}

A query clause typically has this structure:

{

QUERY\_NAME:

{

ARGUMENT: VALUE,

ARGUMENT:VALUE,...

}

}

### Kibana

Kibana is an open-source (Apache Licensed) visual interface for Elasticsearch that works in the browser. Kibana is a snap to setup and start using. It is pretty good at visualizing data stored in Elasticsearch. Written entirely in HTML and JavaScript it requires only a plain webserver. Kibana requires no fancy server side components. The best thing about the Kibana is visualizations are configured completely through the interface and there is no need of program every modification.

In this section, we will be explaining about configuration of Kibana to point at a single index instead of all indices, how to import the data, display the data elegantly and meaningfully on dashboard and searching the data.

## Logstash

Logstash is an open source data collection engine with real-time pipelining capabilities. Logstash can dynamically unify data from disparate sources and normalize the data into destinations of your choice. Cleanse and democratize all your data for diverse advanced downstream analytics and visualization use cases.

While Logstash originally drove innovation in log collection, its capabilities extend well beyond that use case. Any type of event can be enriched and transformed with a broad array of input, filter, and output plugins, with many native codecs further simplifying the ingestion process. Logstash accelerates your insights by harnessing a greater volume and variety of data.

# Chapter 5: Data Analytics

In this information age, information leads to power and success. Using sophisticated technologies such as computers, satellites, etc., we have been collecting tremendous amounts of information. Unfortunately, these massive collections of data stored on disparate structures very rapidly became overwhelming. Today, we have far more information than we can handle: from business transactions and scientific data, to satellite pictures, text reports and military intelligence. Information retrieval is simply not enough anymore for decision-making. Confronted with huge collections of data, we have now created new needs to help us make better managerial choices. These needs are automatic summarization of data, extraction of the "essence" of information stored, and the discovery of patterns in raw data. These needs can be included in the new vertical of Information Technology industry i.e. Data Science.

We, in this project, to help Smart city govern better and interact with the citizens in a much better way, using Data Analytics. Smart city as a whole is a vast area to work on. In particular, we are working on the few use cases listed below.

## 5.1 Defining the use-cases

As far as city governance is concerned, the main issues are related with the water, land, population, and waste. We are working on the optimization of governance in these fields. Our ideas for each department and for a particular use-cases are:

* Waste Management (Garbage Collection)
* Monitoring Gender Equality
* Water Management (Water Supply Requirement)
* Tax Collection (Property Tax collection)

We will discuss each of the use cases one-by-one here.

### 5.1.1 Waste Management (Garbage Collection):

In collaboration with the Health and Hospital Department of Surat Municipal Corporation (SMC), we are working on day-to-day garbage collection.

Waste management or Waste disposal is all the activities and actions required to manage waste from its inception to its final disposal. This includes amongst other things, collection, transport, treatment and disposal of waste together with monitoring and regulation. It also encompasses the legal and regulatory framework that relates to waste management encompassing guidance on recycling etc. The term normally relates to all kinds of waste, whether generated during the extraction of raw materials, the processing of raw materials into intermediate and final products, the consumption of final products, or other human activities, including municipal (residential, institutional, commercial), agricultural, and social (health care, household hazardous waste, sewage sludge).

Waste management is intended to reduce adverse effects of waste on health, the environment or aesthetics. Therefore we have picked the data of various waste generated and manpower and machines required for their disposal at the dumping site for better management of waste disposal in overall city functioning and ensuring healthy neighborhood of the citizens.

Out of the whole garbage collection data, there were fields like door to door garbage collection routes, tons of garbage collection via door to door collection, solid waste disposal, nuisance points attended by the garbage collection team, number of vehicles used for the collection of garbage, number of containers used for the collection of garbage, different administrative charges, etc.

The data was partitioned on the basis of each zones of Surat. After brainstorming within the team and with an expert from SMC, we decided to work on door-to-door garbage collection and number of containers required in each zone in each season in order to maximize the utilization of the manpower and vehicles all over the city.

### 5.1.2 Monitoring gender equality:

As of 2017, gender equality is the 5th of 17th sustainable development goals of the United Nations. Gender equality is measured annually by the United Nations Development Program’s Human Development Reports. Promoting gender equality is seen as an encouragement to greater economic prosperity. Female economic activity is a common measure of gender equality in an economy.

Our idea for this use case is to monitor gender ratio in each zone in comparison with overall Surat city. And hence identify the performance of each zone as far as gender ratio is concerned. For this we have taken census data of 2011 as the base data and zone-wise birth-death data given by Health Department of SMC.

By identifying the performance of each zone for gender equality, we can run awareness campaigns in areas which has lower female-to-male ratio. Sometimes, we can even increase the scrutiny by the health officers in the maternity homes of that area to check whether ill practices are being carried out in those homes or not.

### Water Management (Water Supply Requirement):

Water management is the activity of planning, developing, distributing and optimum use of water resources under defined water policies and regulations. It includes: management of water treatment of drinking water, industrial water, sewage or wastewater, management of water resources, management of flood protection**.** It includes various functions such as water conservation, water resource management, water supply, water treatment.

We worked on the data of water supplied to the whole Surat city on a daily basis. With this data, we can analyze the water requirements of the city and ultimately control the usage of water resources efficiently as we would have clear idea of the water usage in the coming months. In the difficult times like draught and flood, we can hold awareness campaigns to use water smartly.

In collaboration with Water and Irrigation department of SMC, we work on water consumption requirement and based on that predicting the water supply required in order to give better facilities to the citizens.

### 5.1.4 Revenue Collection (Property Tax collection):

Revenue collection is one of the most challenging tasks that falls under the city municipality. Working with Revenue department of SMC, we are planning to make a model such that we could be able to predict the manpower and other resources required for each zone in a particular month time of the financial year. This use case is based on different approach people take to submit their annual property taxes. Revenue department mainly concerns with property tax, professional tax, Octroi and other miscellaneous taxes. We, in this project have worked on property tax collection in the different times of the year.

A property tax (or millage tax) is a levy on the value of a property. The tax is levied by the governing authority of the jurisdiction in which the property is located. This can be a national government, a federated state, a county or geographical region or a municipal. Multiple jurisdictions may tax the same property. This is in contrast to a rent and mortgage tax, which is based on a percentage of the rent or mortgage value. The four broad types of property taxes are land, improvements to land (immovable man-made objects, such as buildings), personal property (movable man-made objects) and intangible property. Real property (also called real estate or realty) is the combination of land and improvements.

Under a property-tax system, the government requires or performs an appraisal of the monetary value of each property, and tax is assessed in proportion to that value.

Typically tax payment duration is divided into three sections: advance bill payment, in bill period bill payment or after bill period payment. This use case is based on different approaches people take to pay their annual property taxes. The pattern of tax payment in particular zone may depend upon the literacy rate, annual income of that zone, number of the households present which could be either occupied or vacant. Hence based on all these attributes we can predict the tax payment nature of the citizens in the particular zone so as to use the office workers more efficiently. This data can also be used to spread awareness in those zones where number of after bill period payments are considerably high so that citizens will pay their dues in time and can avoid late payment penalties.

## 5.2 An architecture of Data Analytics

We have understood the use cases, let’s understand the architecture of data analytics for one use case i.e. garbage collection. There are total 5 steps as far as our garbage collection dataset is concerned. Each step is listed below and explained in sections from here.

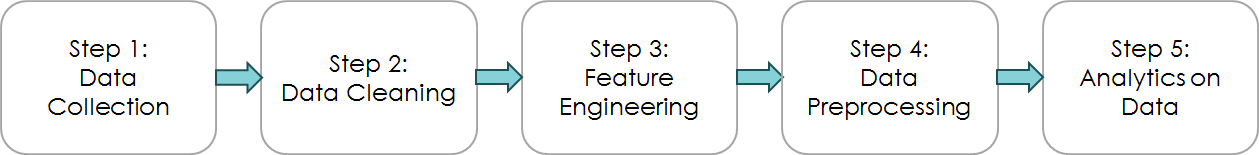
Step 1: Data Collection (associated with the collection of data)

Step 2: Data Cleaning (associated in making data cleaner to use)

Step 3: Feature Engineering (associated with the deriving features to predict)

Step 4: Data Pre-processing (associated in making data more consistent)

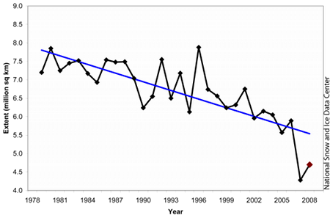
Step 5: Analytics on Data (associated with prediction and inference)



**Figure 9: Architecture of Data Analytics**

## 5.3 Data Collection

Data collection is the process of capturing and scraping data and putting it into some sort of data mart. This data mart could be a Data Warehouse or even a Database. As far as our problem with garbage collection from different zone is concerned, we have used time series database. Time-series databases contain time related data such as logged activities or stock market data. These databases usually have a continuous flow of new data coming in, which sometimes causes the need for a challenging real time analysis. Data mining in such databases commonly includes the study of trends and correlations between evolutions of different variables, as well as the prediction of trends and movements of the variables in time. Figure shows some examples of time-series data.



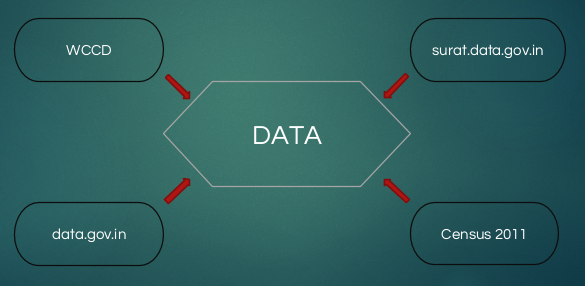
**Figure 10: An example of time series data [21]**

The sources of each of the data attribute is shown in the following table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr No.** | **Data Attribute** | **Description** | **Source** |
| 1. | Zone Area | The area covered by each zone within the city limit. | Zone wise area  (file name: Census\_surat\_households.csv) |
| 2. | Population | The total population residing in each zone from birth and death ratio and the available census data of 2011. | Census data with birth and death numbers. (filename:  Census\_surat\_households.csv, birth\_registration\_20XX.csv, death\_registration\_20XX.csv) |
| 3. | Residential and Non-residential properties | Total residential houses and commercial complexes present in the zone. | Zone wise property count  (file name: Res\_NonRes\_Property\_Count.csv,  Prof\_Tax\_RC\_reg.csv) |
| 4. | Total garbage collected (in Mt.) | The total amount of garbage collected in the containers in each zone. | Collected garbage data  (file name: Health\_Data.csv) |
| 5. | Total garbage collected (in containers) | The number of garbage containers collected in each zone. | Collected garbage data  (file name: Health\_Data.csv) |
| 6. | Female-to-total population ratio | Ratio of female to total population for each zone of the city | (Calculated, surat.data.gov.in, file name : census\_2011) |
| 7. | Total generated bills | Bills generated annually by SMC | Receipt\_Paid\_Count.xlsx, WCCD |
| 8. | Income | Average per capita income of city | Census 2011 |
| 9. | Literacy | Literacy rate of the city | Census 2011 |
| 10. | Wholly, vacant and total households | No. of households present in the city’s area | Surat.data.gov.in  Census\_data\_2011\_households.csv |
| 11. | Working age group  (15-59 years) | Percentage of working age group from total population | WCCD |
| 12. | Advance paid, On-time paid and late paid | Types and counts of bills paid by the citizens | Receipt\_Paid\_Count.xlsx |
| 13. | Water Supply (in MLD) | Water Supplied by various supply pipelines by the municipality.  Data taken from each day of CM’s dashboard | NIC\_details.csv (Surat.data.gov.in) |

**Figure 11: Sources of data attributes [22] [23] [24] [25]**

There are 4 major data sources as far as our smart city project is concerned. These sources are shown in the figure drawn below.



**Figure 12: Major Data Sources**

## 5.4 Data Cleaning

Data cleaning also known as data cleansing and data scrubbing is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. [4]

Real world data tend to be incomplete, noisy and inconsistent. Administratively, incorrect or inconsistent data can lead to false conclusions and misdirected investments on both public and private scales. For instance, the government may want to analyze population census figures to decide which regions require further spending and investment on infrastructure and services. In this case, it will be important to have access to reliable data to avoid erroneous fiscal decisions. In the business world, incorrect data can be costly. Many companies use customer information databases that record data like contact information, addresses, and preferences. For instance, if the addresses are inconsistent, the company will suffer the cost of resending mail or even losing customers.

Hence in our analysis we have cleaned the data by removing some of the outliers present in the data. Some of the values in the given data file was just for the test case purposes or from the range other than we are working on and it was not consistent. Hence to make our model more effective, we have removed these sets of data.

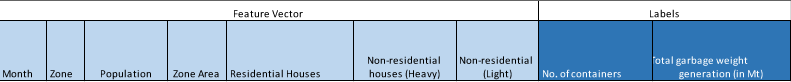
## 5.5 Feature Engineering

Let’s first understand the definition of a feature before diving deeper into feature engineering. In machine learning and pattern recognition, a feature is an individual measurable property of a phenomenon being observed. Choosing informative, discriminating and independent features is a crucial step for effective algorithms in pattern recognition, classification and regression.

### 5.5.1 Garbage System

As far as garbage collection system is concerned, the data which was given to us by SMC had multiple columns such as night scraping number of route, number of people required for night scraping, total garbage collection, door to door garbage collection and containers required for garbage collection, amount generated in (Metric ton) and many more. These data were arranged zone wise with a particular date associated with it.

To start with the garbage collection problem, we decided to take container required for garbage collection as a label class value and try to build the model which will predict the above mentioned class for particular month. Here, the number of containers required for garbage collection is more prominently depend upon the amount of garbage is being generated in the particular zone. Based on this fact, the features affecting the garbage generation would be:



**Figure 13: Feature Vector - Garbage collection**

**Season:** Based on the date, we can predict the nature of garbage generation for particular month. Hence the date plays a crucial role in this model. For example, it has been noted that, in the season of mangos or in summer, the amount of garbage generation is higher in comparison with other seasons.

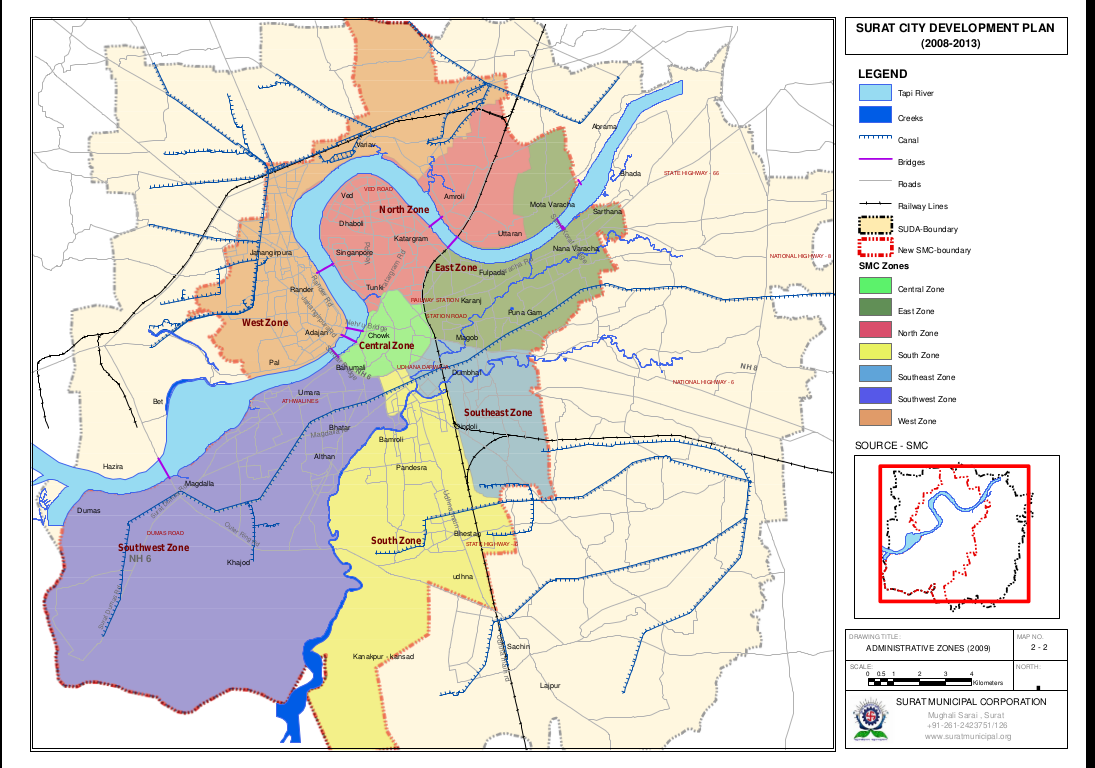
**Zone:** Surat is mainly divided in the seven major zones and they are basically East Zone (EZ), West Zone (WZ), North Zone (NZ), South Zone (SZ), South East Zone (LZ) and South West Zone (AZ).

**Area of the Zone:** Each zone is not divided equally and hence areas are different for each zone which will have different population and hence garbage generation would depend on that.

**Population:** Number of people residing in the particular zone. Greater the number, chances of garbage generation is more.

**Residential and Non-residential houses:** All the buildings and societies are included in residential area whereas all industries and offices which can play role in garbage generation are included in non-residential area.

There can be many more features which can be added to this set of features but currently on the basis of data available we have considered these sets of features only in our experiment.



**Figure 14: Surat divided into 7 zones**

**No. of containers:** Number of containers deployed for the collection of garbage in each area.

**Total garbage collection (in Mt):** Total garbage collection through door-to-door garbage collection teams.

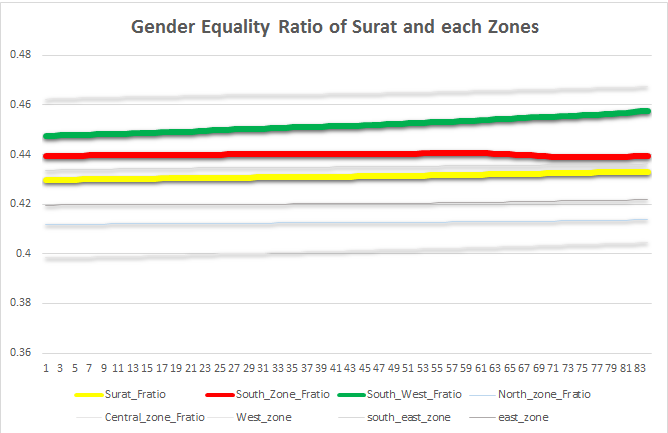
### 5.5.2 Gender Equality

To improve the stability of the society, we have worked on gender inequality issue. For this, we have taken female-to-male ratio over the period of 7 years from 2009 to 2015.

****

**Figure 15: Feature Vector - Gender Equality**

Knowing this data from birth and death rate of each zone and the base male and female population from census 2011’s data, we have analyzed the performance of gender ratio over the period. As shown in the graph below, the data series plotted in yellow color is overall gender equality performance of Surat city. Whereas, the data series shown in green color is South West Zone. Which shows that it has the best performance in comparison with all other zones. Here the best performance means the improvement in gender equality ratio. The data series shown in red color is South Zone whose performance is the worst in compare to other zones.



**Figure 16: Gender Equality Ratio of Surat and each Zones**

### 5.5.3 Water Supply Management

Water supplied to city depends on various feature such as its population and also depending whether the number of residential house or non-residential houses of the city. Data given to us had amount of supplied water in million liters of each date from year 2015 to 2017. Features of the water supply analysis is shown in the below table:



**Figure 17: Feature Vector - Water Supply**

**Date:** The first two columns contain month and day of the supplied water.

**Population:** The amount of water supplied will definitely depend on the number of citizens using the resources and hence date population is recorded.

**Residential and Non-Residential Properties:** Residential properties tend to use more water and hence city with more residential areas will require more water.

### 5.5.4 Property Tax Collection

Basic aim of this use cases was to predict the count of bill which is paid in advance, in bill period or after bill period. The pattern of tax payment depends upon the behavioral nature and also the attitude of citizens towards tax which can be predicted with following features:

****

**Figure 18: Feature Vector – Property Tax Collection**

**Zone:** Zone has same notation as described in waste management.

**Total Payable bills:** It gives the count of total property bills that are to be paid in any particular zone in a particular zone.

**Total paid bills:** It contains the bills that are actually paid by the citizens. There are some citizen who does not pay their taxes and hence bills are generated but not paid which can be estimated with this column.

**Income:** Property tax of any particular zone will marginally depend on the capita income of that zone in that year. Hence it also plays important role in the pattern of tax payment.

**Literacy Rate:** Literate people tend to pay their well in advance or otherwise in bill period time. Hence pattern of tax payment also depend upon the literacy of the zone also.

**Wholly, Vacant, Total households:** Wholly, vacant and total number of households present in the city’s defined area.

## 5.6 Data Preprocessing

Data preprocessing is a data mining technique that involves transforming raw data into an understandable format. Real-world data is often incomplete, inconsistent, and/or lacking in certain behaviors or trends, and is likely to contain many errors. Data preprocessing is a proven method of resolving such issues. Data preprocessing prepares raw data for further processing. [4]

Before data mining algorithms can be used, a target data set must be assembled. As data mining can only uncover patterns actually present in the data, the target data set must be large enough to contain these patterns while remaining concise enough to be mined within an acceptable time limit.  Pre-processing is essential to analyze the multivariate data sets before data mining. The target set is then cleaned. Data cleaning removes the observations containing noise and those with missing data.

Data preprocessing includes resolving conflicts within the data, normalization and generalization of data, reduction of a number of values present in dataset. As far as data preprocessing is concerned, we have done three major modifications in our dataset in order to use it in analytics section.

**Calculating population month-wise:** We had population of each zone based on the census 2011. We also had the data of number of births and deaths occurred in each zone month wise from 2009 to 2015. Hence based on this we calculated the population of each month and associated that with each zone.

**Standardized Zone Code:** In the datasets we have, there was not any standardized notation for representing the zones. For example, some of the datasets have used the notation of Zone name while the others have used Zone Code or Area Name as mentioned in the table below. Hence to clean the data and generalize it, we have used one standardized code for each zone for the analysis purpose.

|  |  |  |  |
| --- | --- | --- | --- |
| Zone Name | Zone Code | Area Name | Standardized Code for Analysis |
| East Zone | EZ | Varachha | 0 |
| Central Zone | CZ | - | 1 |
| North Zone | NZ | Katargam | 2 |
| South Zone | SZ | Udhana | 3 |
| South East Zone | SEZ, LZ | Limbayat | 4 |
| South West Zone | SWZ, AZ | Athwa | 5 |
| West Zone | WZ | Rander | 6 |

**Figure 19: Data Preprocessing - Standardization of zone codes**

**Separation of each parameters from date:** Similarly to make date format standardized, we extracted date, month and year separately from the timestamp.

|  |  |  |  |
| --- | --- | --- | --- |
| Date-stamp | Year | Month | Date |
| 02/11/2013 | 2013 | 11 | 02 |

**Figure 20: Data Preprocessing - Separation of date from date-stamp**

We represented month as a number ranges as explained in the table below.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Jan | Feb | March | April | May | June | July | Aug | Sept | Oct | Nov | Dec |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

**Figure 21: Data Preprocessing - Representation of Months**

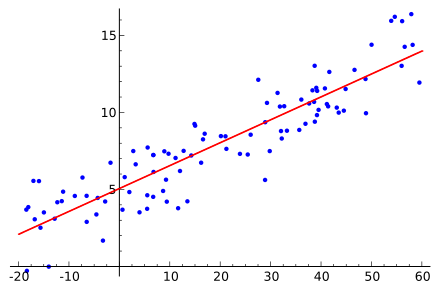
## 5.6 Analytics on Data

Once we have transformed our data into a desirable format, we perform analytics over this dataset in order to examine it and draw conclusions about the information it contains. Among all the data analytics techniques available at our disposal, we decided to proceed with linear regression, Gaussian process and SVM classifier.

**Linear Regression:** In statistics, linear regression is an approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X. The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression.

In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models. Most commonly, the conditional mean of y given the value of X is assumed to be an affine function of X; less commonly, the median or some other quartile of the conditional distribution of y given X is expressed as a linear function of X. Like all forms of regression analysis, linear regression focuses on the conditional probability distribution of y given X, rather than on the joint probability distribution of y and X, which is the domain of multivariate analysis.

For our application, linear regression has been used to fit a predictive model to an observed data set of *y* and *X* values. After developing this model, additional values of *X* is then given without their accompanying value of *y*, and the fitted model is used to make a prediction of the value of *y*.



**Figure 22: Random data points and linear regression**

**Gaussian Process:**

A Gaussian process uses lazy learning and a measure of the similarity between points (the kernel function) to predict the value for an unseen point from training data. The prediction is not just an estimate for that point, but also has uncertainty information—it is a one-dimensional Gaussian distribution (which is the marginal distribution at that point).

Gaussian is based on the notation of the Gaussian distribution (normal distribution). Gaussian processes can be seen as an infinite-dimensional generalization of multivariate normal distribution.

**Support Vector Machine:**

Support Vector Machine (SVM aka Support Vector Networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

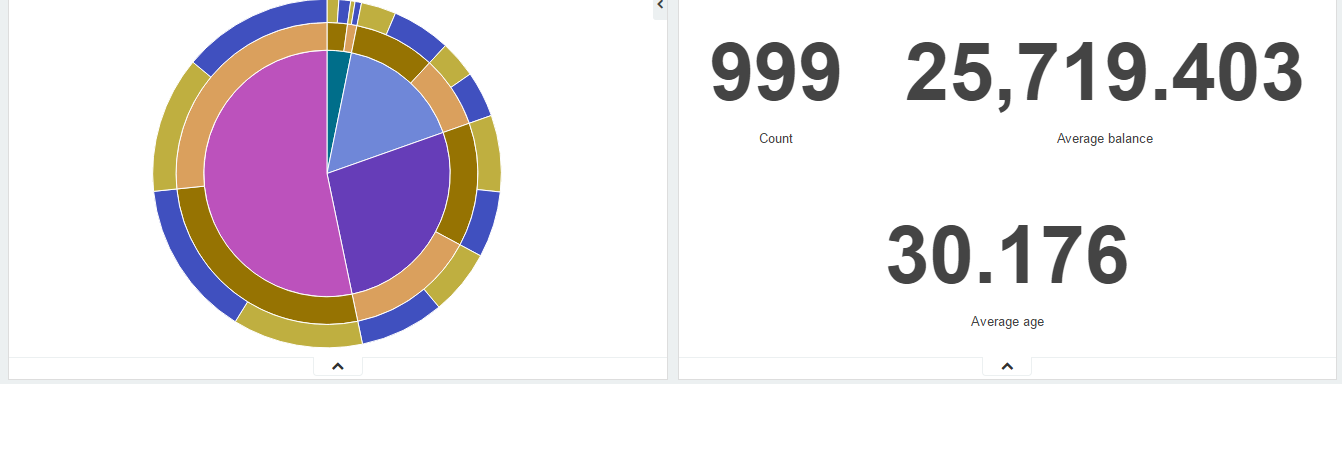


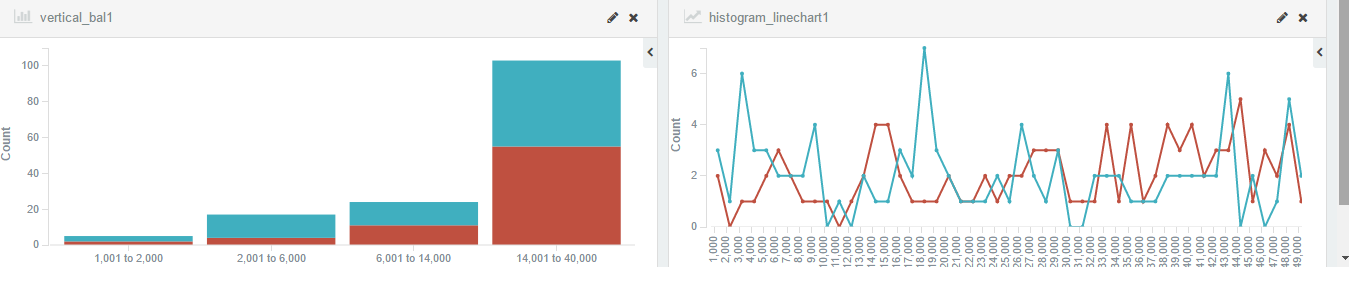
**Figure 23: Support Vector Machine**

# Chapter 6: Practical Implementation

We’ve worked on mainly two fields for enhancing the data analytics platform- visualization-and-analytics and prediction.

For visualization and analytics, we have used ELK Stack (Elasticsaerch, Logistash and Kibana). We have worked on Elasticsearch v2.4.1, Kibana v4.6.1 and Logstash v5.0 on windows10 operating system. Sense is a browser plugin for communicating with Elasticsearch. Initially, a localhost is set and a node is created on cluster. To start with, we developed a dashbaord for a sample polutants data collected via IoT sensors. The dashboard can be accessed from any of the machine connected to the internet. The created dashboard is shown in the Figure-9.





**Figure 24: Developed dashboard for a polutants data**

For prediction, we have implemented linear regression classifier, Gaussian classifier and SVM classifier in MATLAB for all the use cases. We have used the Garbage collection data and the Census dataset provided by Surat Municipal Corporation (SMC) for the various zones of Surat.

We have considered two different labels viz. number of containers utilized and total garbage weight in Mt.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | (MEAN = 2792) | Root Mean Square Error (RMS) | | |
| No. | Features present in the feature vector | Linear | Gaussian | SVM |
| 1. | 2 (year, month) | 248.06 | 249.12 | - |
| 2. | 3 (zone, population, area) | 67.83 | 377.10 | 61.42 |
| 3. | 3 (residential, non-residential) | 629.31 | 413.95 | 1384.00 |
| 4. | 5 (year, month, zone, population, area) | 61.98 | 279.91 | 4327.70 |
| 5. | 5 (year, month, residential, non-residential-light, non-residential-heavy) | 21.29 | 394.15 | 925.37 |
| 6. | 6 (zone, population, area, residential, non-residential-heavy, non-residential-light) | 26.88 | 394.15 | 742.38 |
| 7. | 6 (year, month, zone, population, area, residential, non-residential-heavy, non-residential-light) | 57.91 | 389.52 | 231.79 |

**Figure 25: Performance table - Garbage Collection - No. of containers**

Now, the feature set has been selected in such a way that the error is decreased as has been shown in the below table. Prediction for the two labels was carried on for the dataset by adding features as shown in the table above.

|  |  |  |
| --- | --- | --- |
|  | (MEAN = 6119) | Root Mean Square Error (RMS) |
| No. | Features present in the feature vector | Linear |
| 1. | 2 (year, month) | 2796.7 |
| 2. | 3 (zone, population, area) | 1528.4 |
| 3. | 3 (residential, non-residential) | 4447.6 |
| 4. | 5 (year, month, zone, population, area) | 926.7 |
| 5. | 5 (year, month, residential, non-residential-light, non-residential-heavy) | 1146.6 |
| 6. | 6 (zone, population, area, residential, non-residential-heavy, non-residential-light) | 521.0 |
| 7. | 6 (year, month, zone, population, area, residential, non-residential-heavy, non-residential-light) | 561.4 |

**Figure 26: Performance table - Garbage Collection - Weight collected (in Mt.)**

Gender Equality datasets can be summarized as follows. South Zone has the best performance and the South West has the worst as far as gender equality is concerned.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Female to Total Population ratio | |  |  |
| Zone \ Year | 2009 | 2015 | Net Change  (in percentage) | Remarks |
| East Zone | 0.419633096 | 0.421884579 | 0.2251484 |  |
| Central Zone | 0.462339642 | 0.467306818 | 0.4967176 |  |
| North Zone | 0.411810745 | 0.413682224 | 0.1871479 |  |
| South Zone | 0.43945633 | 0.43934257 | -0.011376 | Worst Performance |
| South East Zone | 0.433693533 | 0.436195402 | 0.2501869 |  |
| South West Zone | 0.447492138 | 0.457388667 | 0.989653 | Best Performance |
| West Zone | 0.398304312 | 0.404454107 | 0.6149795 |  |
| Surat City | 0.429718226 | 0.433107099 | 0.3388874 | Overall city Performance |

**Figure 27: Data Analytics - Gender Equality Performance (City and all zones)**

Property Tax Collection results for linear regression, Gaussian process and SVM process are shown below in the table.

|  |  |  |  |
| --- | --- | --- | --- |
|  | (MEAN = 20643) | Root Mean Square Error (RMS) | |
| No. | Features present in the feature vector | Linear | Gaussian |
| 1. | 1 (zone) | 550.0 | 1322.3 |
| 2. | 3 (Zone, paid bills, generated bills) | 3216.8 | 1322.3 |
| 3. | 6 (zone, paid bills, generated bills, income, age-group, literacy) | 1704.3 | 1322.3 |
| 4. | 9 (zone, paid bills, total generated bills, income, age-group, literacy, wholly households, vacant households, total households) | 51.4 | 13025.3 |

**Figure 28: Data Analytics - Property Tax (Advance Payment)**

|  |  |  |  |
| --- | --- | --- | --- |
|  | (MEAN = 54599) | Root Mean Square Error (RMS) | |
| No. | Features present in the feature vector | Linear | Gaussian |
| 1. | 1 (zone) | 4946.1 | 4783.6 |
| 2. | 3 (Zone, paid bills, generated bills) | 4403.0 | 1146.7 |
| 3. | 6 (zone, paid bills, generated bills, income, age-group, literacy) | 4443.1 | 572.0 |
| 4. | 9 (zone, paid bills, total generated bills, income, age-group, literacy, wholly households, vacant households, total households) | 3420.8 | 2799.6 |

**Figure 29: Data Analytics - Property Tax (On-time payment)**

|  |  |  |
| --- | --- | --- |
|  | (MEAN = 87192) | Root Mean Square Error (RMS) |
| No. | Features present in the feature vector | Linear |
| 1. | 1 (zone) | - |
| 2. | 3 (Zone, paid bills, generated bills) | 1186.3 |
| 3. | 6 (zone, paid bills, generated bills, income, age-group, literacy) | 6147.4 |
| 4. | 9 (zone, paid bills, total generated bills, income, age-group, literacy, wholly households, vacant households, total households) | 3369.3 |

**Figure 30: Data Analytics - Property Tax (Late payment)**

Water Supply Management and prediction accuracy in terms of Root Mean Square error of supplied water each data is shown in the table below.

|  |  |  |  |
| --- | --- | --- | --- |
|  | (MEAN = 1088) | Root Mean Square Error (RMS) | |
| No. | Features present in the feature vector | Linear | Gaussian |
| 1. | 1 (population) | 10.55 | 65.79 |
| 2. | 2 (residential, non-residential) | 31.88 | 63.54 |
| 3. | 2 (month, day) | 10.36 | 98.65 |
| 4. | 3 (population, residential, non-residential) | 53.82 | 63.54 |
| 5. | 3 (month, day, population) | 39.85 | 63.39 |
| 6. | 4 (month, day, population, residential, non-residential) | 43.97 | 66.42 |

Hence, we were successful in making our model more and more efficient by adding more and more relevant features.

# Chapter 7: Conclusion

In the very beginning of this academic year in June 2016, we were given a set of requirements by SMC officials to work on. Their requirements were changing over the period of time after more and more brainstorming. So, we stopped at one point and started working on the set of problems.

Throughout the year, we have worked on various datasets and tried to make more and more efficient models for each and every problem. After the thorough analysis of each data on various classifiers, we have picked a set of attributes and one best model for each datasets. The results are quite impressive and we are proposing these techniques to SMC.

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. | Dataset | Feature Vector | Regression Model |
| 1. | Garbage Collection | Month, Zone, Population, Zone Area, Residential Houses, Non-residential properties (heavy), Non-residential properties (light) | Linear Regression Model |
| 2. | Water Supply Management | Month, Day, Population | Linear Regression Model |
| 3. | Property Tax Collection | Zone, Total Paid bills, Total generated bills, Income (in thousands), Working Age group (15-59 years in percentage), Literacy, Wholly households, vacant households and total households  Zone, Total Paid bills, Total generated bills, Income, Working Age group (15-59 years in percentage), Literacy | Linear Regression Model  Gaussian Process |

# Chapter 8: Future Work

So far, we have analyzed various components required to build a smart city. The challenge of getting useful data from the unstructured data is overcome by Elasticsearch, which is a multi-tenant search engine. To make this data more human friendly, we have used Kibana for visualization. We have also predicted number of containers required for garbage collection with the derived features with the help of Matlab.

In future, this work can be extended by implementing more use-cases based on the data available and requirements specified by the Surat Municipal Corporation. Till now, we have only used approximately 10 features according to the availability of data, but in future we can incorporate more features and make our existing models more efficient and accurate in predicting the labels. Even various other classifiers and regression models can be used for each of the use cases. For the specific problems, we can implement ensemble classifiers to take the advantages of two or more models. Thus, we will work on make governance simplified for the selected municipality departments.

# References

1. 7 uses of analytics in smart cities <https://channels.theinnovationenterprise.com/articles/158-7-uses-for-analytics-in-smart-cities>
2. Data and Analytics: Resources for Smart Cities, A report by the Data and Analytics Task Group for the Smart Cities Forum <https://connect.innovateuk.org/documents/6697210/0/SCF_Data+and+Analytics_GOS.pdf/d35b1135-acb5-4222-8dce-a2ac39bcfa04?version=1.0>
3. Sustainable and smart cities <http://algoritmi.uminho.pt/ts-cities/objectives/>
4. Wikipedia, the free encyclopedia.
5. Smart Cities Challenge, Ministry of Urban Development, Government of India <http://www.smartcitieschallenge.in/>
6. Smart Cities: Background paper, Government of United Kingdom <https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/246019/bis-13-1209-smart-cities-background-paper-digital.pdf>
7. Issues Paper On Smart Cities and Infrastructure, United Nations Commission on Science and Technology for Development <http://unctad.org/meetings/en/SessionalDocuments/CSTD_2015_Issuespaper_Theme1_SmartCitiesandInfra_en.pdf>
8. Key Performance Indicators of smart sustainable cities, ITU, United Nations agency for Information and communication technology, Singapore. <https://www.itu.int/net/pressoffice/press_releases/2015/43.aspx>
9. Connected Cities: Sustainable Smart Cities <https://itu4u.wordpress.com/2013/09/15/connected-cities-smart-sustainable-cities/>
10. Personal Communications, Rob van den Dam, IBM Institute for Business Value, ITU Telecom World 2013. [http://www.ijltet.org/journal//14686500015.pdf](http://www.ijltet.org/journal/14686500015.pdf)
11. The “Internet of Things (IoT)” things by Kevin Ashton <http://www.rfidjournal.com/articles/view?4986>
12. IoT, IoE, & M2M sector <https://www.abiresearch.com/research/service/internet-of-everything/>
13. Pattern Recognition Techniques on Data Mining <http://www.journals.elsevier.com/pattern-recognition-letters/call-for-papers/special-issue-on-pattern-recognition-techniques-data-mining>
14. Kononenko O, Baysal O, Holmes R, Godfrey MW. Mining modern repositories with elasticsearch. In: Proceedings of the 11th working conference on mining software repositories. Hyderabad, India: ACM; 2014. p. 328–31.
15. Gormley, C., & Tong, Z. (2014). Elasticsearch: The Definitive Guide: O'Reilly Media, Inc.
16. Kibana User Guide: Kibana v4.6.1 <https://www.elastic.co/guide/en/kibana/4.6>
17. Logstash User Guide: Logstash v5.0 <https://www.elastic.co/guide/en/logstash/current/index.html>
18. Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. ArXiv: 1603.04467, 2016.
19. Census Income Data Set: <https://archive.ics.uci.edu/ml/datasets/Census+Income>
20. Principles of Knowledge Discovery in Databases (CMPUT690) by Osmar R. Zaïane. <https://webdocs.cs.ualberta.ca/~zaiane/courses/cmput690/notes/Chapter1/ch1.pdf>
21. Time series dataset <http://www.seos-project.eu/modules/timeseries/timeseries-c05-p06.html>
22. Surat Municipality Online Data open forum <https://surat.data.gov.in/>
23. Indian Open Data Platform <https://data.gov.in>
24. Official Census Data of 2011 <http://www.census2011.co.in/>
25. World Council on City Data (WCCD) <http://open.dataforcities.org>

# Acknowledgement

Behind any major work undertaken by an individual there lies the contribution of the people who helped him to cross all the hurdles and achieve his goal. So, we would like to express our deep gratitude to Dr. Dhiren R. Patel, Professor and Head of Computer Engineering Department, Sardar Vallabhbhai National Institute of Technology aka NIT, Surat for helping & motivating us with his fruitful advice.

We are also very thankful to Jigar Patel, Associate Engineer (Computer) at Surat Municipal Corporation (Information Systems Department) to help us understand Surat smart city’s existing infrastructure and guide us throughout the project and technicalities.