Swami Keshvanand Institute of Technology, Management & Gramothan

Ramnagariya, Jagatpura, Jaipur - 302017

Software Requirement Specification

On

Criminal Activities Prediction In The City Using ML

Department of Computer Science & Engineering

SUBMITTED TO: **SUBMITTED BY:**

Dr. Tapas Badal   Ishan Kalra – 15ESKCS726

(Department of Computer Science   Chandra Prakash – 15ESKCS716

   & Engineering)   Bhumika Saxena – 15ESKCS715

MENTORED BY:

Dr. Basant Agarwal

Associate Professor

(Department of Computer Science &
Engineering

Table of Contents

Introduction

As the population is increasing so is the crime rate. This project is aimed at solving this using modern data analysis and machine learning techniques.

1.1   Application Areas

This project can have wide implications from efficient use of resources and man power to reduction of criminal events in the city.

1.2   Problem statement

To analyze past criminal data and generate actionable insights and also use machine learning to predict future criminal events.

1.3   Technical domain

The project under the domain of data analysis and machine learning.

1.4   Existing methods

Many countries are investing in machine learning and AI to devise solutions to combat crime.

1.5   Methodology

The project is aimed at analyzing past criminal data and predict future criminal events and also gathering insights from the data.
The data will be sourced from Boston Police Criminal Datasets – Analyze Boston.
Data Analysis and Data visualization techniques will be used to gather insights from the data and then Machine Learning techniques will be used to predict future criminal events.
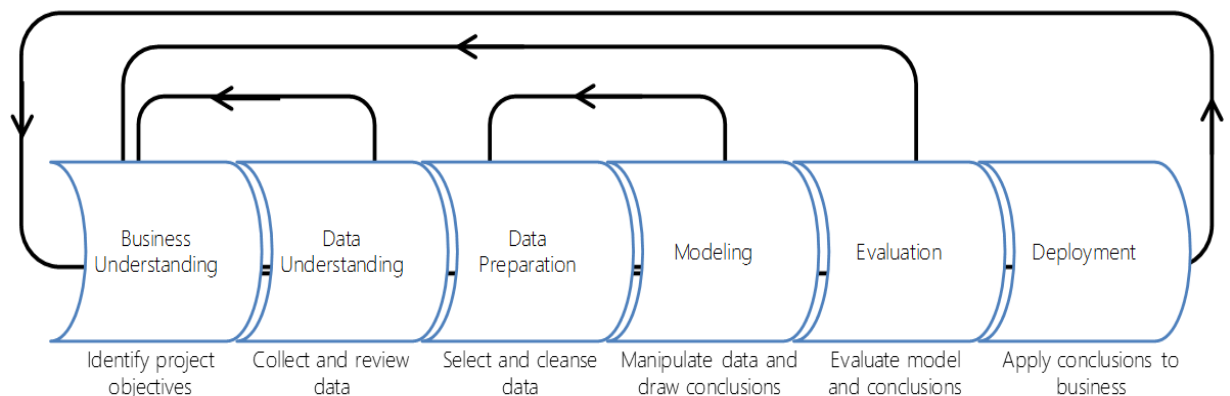
The proposed workflow of the project will be as follows:
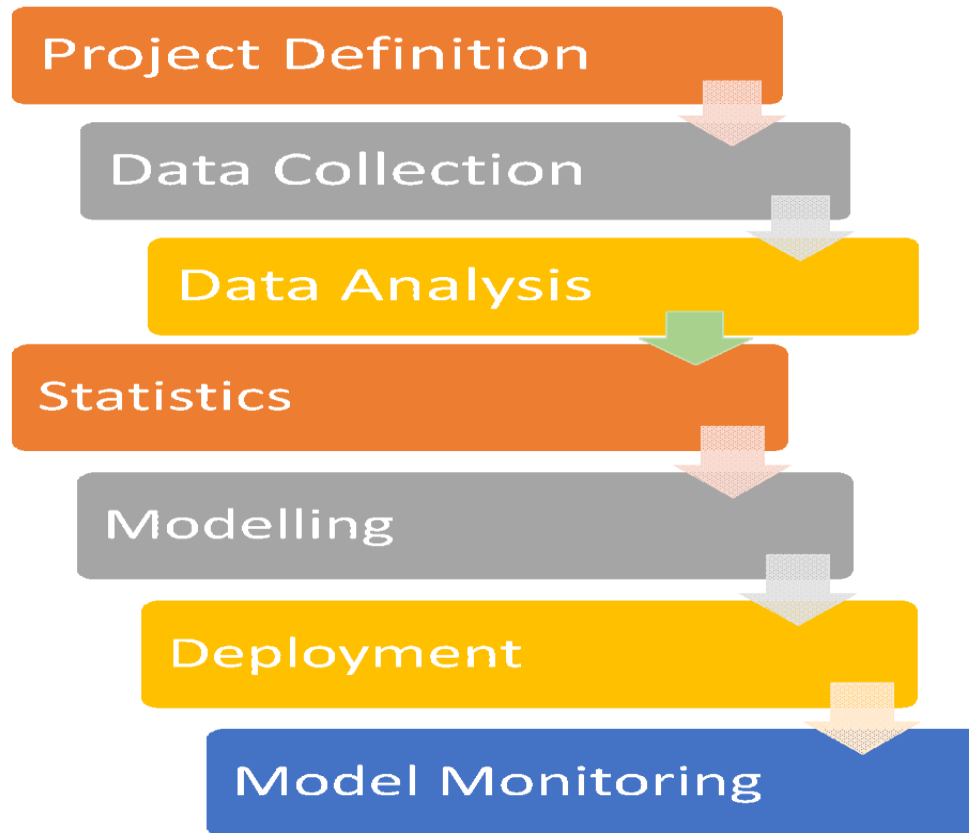
Figure 1.1 Proposed work-flow of the project.

We are following **C**ross **I**ndustry **S**tandard for **D**ata **M**ining (**CRISP-DM**)
Which is an iterative model for developing and deploying machine learning projects.

Figure 1.2 CRISP-DM model for machine learning projects.

1.6   Purpose of the project

As we know rapid population explosion throughout the world is causing major problems for the government and authorities as they are unable to their resources problem. One serious consequence of this rapid population explosion is increase in number of criminal events. For several years authorities plan to combat criminal activities by manually analyzing the data and taking decisions. As number of criminal cases have increased over the year this manual planning and deployment of resources have become inefficient. Our aim is to collect the criminal data from the authorities, analyze the data and gathering valuable insights and pattern from the data. Further we will use machine learning techniques to predict the occurrences of future criminal events.

1.7   Scope

- The project aims at automating the process of analyzing and visualizing criminal data.
- The project can be extended for other locations as long as data is consistent with the format supplied.
- The project will significantly help the authorities in planning and deployment of man power and other resources.

Figure 1.3 Illustration picture

1.8   Tools Used

The tools and technologies which are used in this project are:

1.8.1   Application & Architecture

- **Python**

  Python is an interpreted high-level programming language for general-purpose programming. Python provides many APIs for data analysis and machine learning and the support for it is continuously growing. IPython is used to develop this project because of its increased efficiency and added features.

- **JavaScript**

  JavaScript is a programming language that allows you to implement complex things on web pages – every time a web page does more than just sit there and display static information for you to look at – displaying timely content updates or interactive maps, or animated 2D/3D graphics, or selling video jukeboxes, etc. You can bet that JavaScript is probably involved. It is the third layer of the layer cake of standard web technologies, two of which (HTML and CSS) we have covered in much more detail in other parts of the Learning Area.

1.8.2 Packages & Libraries

  Apart from high-level programming languages described above these frameworks and libraries are used in the development of the project.

- **Django**

  Django is a high-level python web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of web development, so you can focus on writing app without needing to reinvent the wheel.

- **Scikit-learn**

  Scikit-learn is a python machine learning library. It features various classification, regression and clustering algorithms including support vector machines.

- **Matplotlib**

  Matplotlib is a python plotting library.

- **Numpy**

  NumPy is python library which stands for numeric python. It has support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on this arrays.

- **Pandas**

  Pandas is a python library for data manipulation and data analysis. In particular it offers data structures and algorithms for manipulating numerical tables and support for time series.

- **Seaborn**

  Seaborn is python library for data visualization and faceting based on matplotlib. It provides high level interface for drawing attractive and informative statistical graphs.

- **FBProphet**

  Prophet is python library for forecasting time series data based on additive model where non-linear trends are fit with different features.

1.8.3 Database Platform

Data base is an integral part of any product. We're using following databases:

- **SQLite**

  SQLite is a self-contained, high-reliability, full featured, public domain, SQL database engine.

1.8.4 Design Tools

Following tools are used in development of this project:

- **RSA – Rational Software Architect**

  IBM Rational Software Architect (RSA) is a comprehensive design, modeling and development tool for end-to-end software delivery. It uses Unified Modeling Language (UML) for designing enterprise applications and web services.

1.9 References

- Object Oriented Modeling and Design with UML – Michael Blaha, James Rambaugh.
- Software Engineering, Seventh Edition, Ian Sommerville.
- IBM RAD Books.
- IBM TGMC Sample Synopsis.
- IBM – www.ibm.com/developerworks
- JavaScript – www.developers.Mozilla.org
- Python3 – www.python.org
- Motivation for the project - https://blog.dataiku.com/predicting-london-crime-rates-using-machine-learning

1.10 Technologies To Be Used

- SQLite – Database Management
- Rational Software Architect
- IDE – Jupyter notebook
- Cloud Environment – Microsoft Azure Machine Learning Studio.
- Additional Visualization Tool – Microsoft Azure PowerBI Desktop.

2 Overview

This section covers a brief overview of the project, our proposed solution other details of the project.

2.1 Existing System

As of 2018, India doesn't have any digital crime analyzing platform and neither a digitized criminal database.
This project is aimed to solve the first problem i.e. criminal activities analysis.

2.1.1 Our Plan

The project will use data analysis and machine learning techniques along with modern tools to analyze criminal activities.

- Our plan is to reduce time required to gather insights from the data.
- Our plan is to build a product which will help authorities in better planning and efficient use of resources.
- To help general public know about the criminal activities in their areas so they can stay aware.

2.2   Software Interface

The following software dependencies are used in development of project:

- **Client on Internet**

  Web browser and operating software (any)

- **Database server**

  SQLite database

- **Development End**

  IDE – Jupyter notebook, Microsoft Azure Machine Learning Studio, Chrome Browser.

2.3   Hardware Interface

The server is directly connected to client systems and client systems have access to the database in the server.
The minimum system requirements is as follows:

Table 2.1 Hardware requirements.

Client Side			
	Processor	RAM	Disk Space
Internet Explorer 6.0	Pentium II at 500 MHz	64 MB	1 GB
Server Side			
Web sphere application server V6.1	Pentium III at 1GHz	512 MB	2 GB
DB2 V8.1	Pentium III at 1GHz	512 MB	1GB (Excluding data size)

User Characteristics:

- o **End User** – This is authority who is intended to use this product for analyzing and gathering actionable insights and see predictions.
- o **Administrator** – This person is responsible for certain types of changes in the code, managing the sources of data to be analyzed.

2.4   Communication Interface

- End user on the internet will be using HTTP/HTTPS Protocols.
- The Administrator on the internet will be using HTTP/HTTPS along with SSL and other protocols.

2.5   Constraints

- The end user can see the generated insights and predictions on computer browsers and later on this functionality will be available for mobile devices as well.
- Although the security is given high importance, there is always a chance of intrusion in the web world which requires constant monitoring.
- The authorities must keep in mind that though the system is trained to generate insights and prediction based on data, there is always a chance that these predictions might not always 100% correct and accuracy levels are provided along with predictions generated.
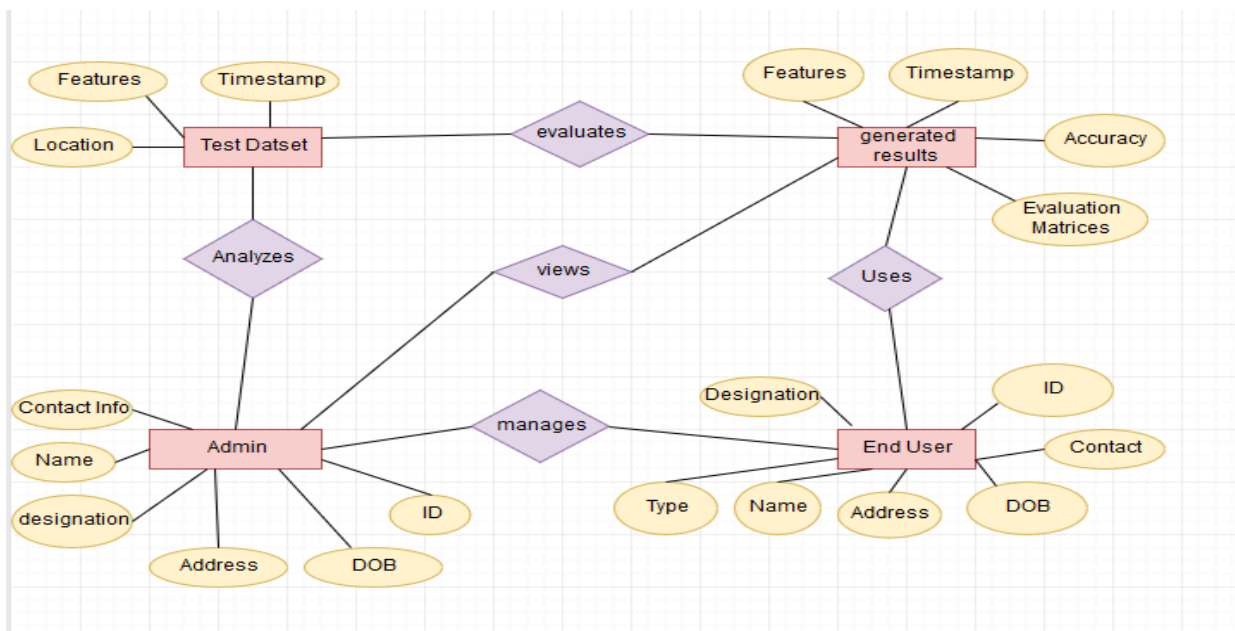
2.6   The E-R Diagram

Figure 2.1 The E-R diagram

Description

Entities

There are four entities which are:

1. Admin: The Attributes are-ID, DOB, Name, Contact, Address, Designation
2. Test Dataset: The attributes are- Features, Timestamp, Location
3. End User: The attributes are-ID, Type, Name, Address, DOB, Contact
4. Generated resu.lts: The attributes are-Features, Timestamp, Accuracy, Evaluation Matrices

Relationships

There are 5 relationships which are:

1. Analyzes: It is between Admin and Test Dataset. The Admin analyzes the test dataset and generate insights.
2. Evaluates: It is between Test Dataset Generated results. When admin analyzes test dataset the insights and predictions are generated.
3. Views: It is between Admin and Generated Results. The admin the generated results by him.
4. Manages: It is between Admin and End User. The admin have all the privileges to add, remove the users.
5. Uses: It is between End User and Generated Results. The End User have rights to see the generated insights and Prediction. The amount of data which an End User can access depends upon type of user.
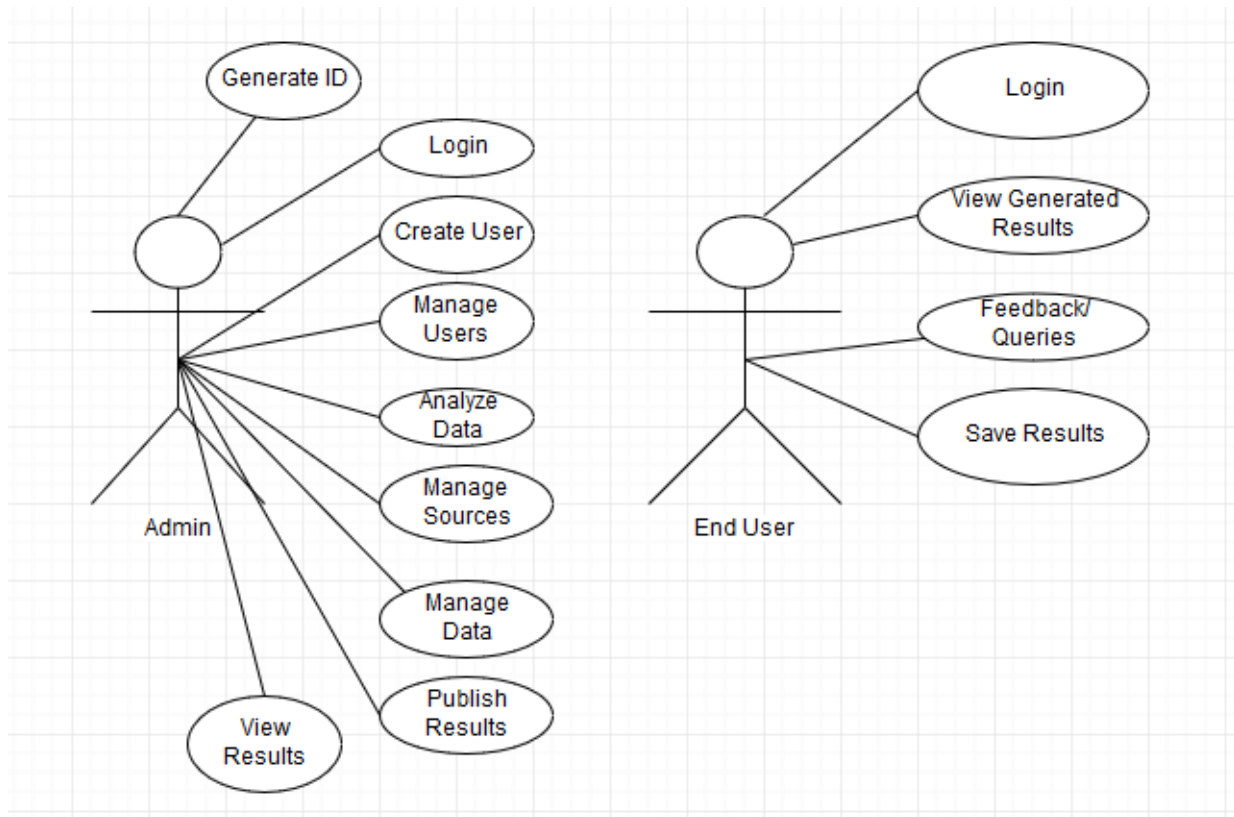
2.7   Use-Case Diagram:

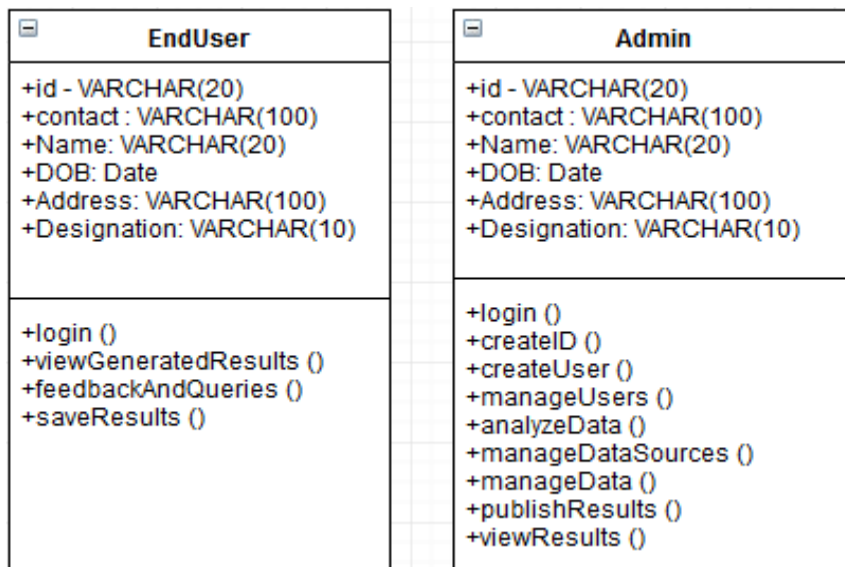*Figure 2.2 use-case diagram for Admin and End User*

Description

Admin: Admin has various responsibilities which are:
- **Login:** Login into the dashboard
- **Create ID:** Generation of IDs for potential users of the product.
- **Create User:** Create an End User.
- **Manage user:** Manage an End User.
- **Analyze Data:** Analyze the data which is being supplied.
- **Manage Sources:** Manage various sources of data i.e. from data base, streaming, csv, stata, etc.
- **Manage Data:** Manage the existing data.
- **Publish Results:** publish the generated insights and results.
- **View Results:** View the results generated by him.

End-User: The responsibilities of end user includes:
- **Login:** The login is required to determine the type of user.
- **View Generated Results:** The user can view generated results and matrices.
- **Save Results:** The user can save results for future use.
- **Feedback/ Queries:** The user can post feedback about results and ask any queries to the admin.

2.8   Class Diagram

Figure 2.3 Class Diagram for Admin and End User

Description

The above class Diagram show all the required classes involved. These are:

Admin
- **Attributes:** Admin has various attributes that includes:
  - **Id:** Admin's ID.
  - **Contact:** Admin's Contact Details.
  - **Name:** Admin's Name
  - **DOB:** Admin's Date of Birth.
  - **Address:** Admin's Address.
  - **Designation:** Admin's Designation.

- **Operations:** Admin can perform following operations:
  o **Login ():** Login into the dashboard.
  o **createUser ():** create a new user.
  o **manageUsers ():** manage existing user.
  o **createID ():** generate ID for new User.
  o **analyzeData ():** Analyze current data.
  o **manageDataSources ():** Manage Data Sources.
  o **manageData ():** Manage Current Data.
  o **publishResults ():** Publish analyzed results.
  o **viewResults ():** View generated results.

End User

- **Attributes:** End User has many attributes that includes:
  o **Id:** End User's ID.
  o **Contact:** End User's Contact Details.
  o **Name:** End User's Name
  o **DOB:** End User's Date of Birth.
  o **Address:** End User's Address.
  o **Designation:** End User's Designation.

- **Operations:** The End User can perform following operations:
  o **Login ():** Login into the dashboard.
  o **viewGeneratedResults ():** View published results.
  o **saveResults ():** save results for later use.
  o **feedbackAndQueries ():** Given Admin feedback or ask queries.