A Major Project

On

**CRIME ANALYSIS AND PREDICTION**

Submitted in partial fulfilment of the requirements for the award of the degree of

**Bachelor of Technology**

in

**Department of Computer Science and Engineering**

Submitted by

|  |  |
| --- | --- |
| **Y. BHARATH** | **15241A05H9** |
| **G. HARI KRISHNA** | **15241A05D2** |
| **UZAIR ADEEL** | **15241A05F0** |
| **J. REVANTH** | **15241A05D5** |
| **K. ROHITH KUMAR** | **16245A0529** |

Under the Esteemed guidance of

**V. SRILAKSHMI**

**Assistant Professor**



**Department of Computer Science and Engineering**

**GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY**

**(Approved by AICTE, Autonomous under JNTUH, Hyderabad)**

**Bachupally, Kukatpally, Hyderabad-500090**

**2018-2019**



GOKARAJU RANGARAJU INSTITUTE OF ENGINEERING AND TECHNOLOGY

**(Approved by AICTE, Autonomous under JNTUH, Hyderabad)**

**Department of Computer Science and Engineering**

# CERTIFICATE

|  |  |
| --- | --- |
| INTERNAL GUIDE | HEAD OF THE DEPARTMENT |
|  |  |
| **V. Srilakshmi** | **Dr. K. MADHAVI** |
| **Assistant Professor** | **Professor** |

This is to certify that the major project titled “**Crime Analysis and Prediction”** is submitted by **Y. Bharath (15241A05H9), G. Hari Krishna (15241A05D2), Uzair Adeel(15241A05F0), J. Revanth (15241A05D5) and K. Rohith Kumar (16245A0529)** inpartial fulfillment of the award of degree in BACHELOR OF TECHNOLOGY in Computer Science and Engineering during academic year 2018-2019.

EXTERNAL EXAMINER

# ACKNOWLEDGEMENT

This is to express our indebted gratitude to our supervisor, **V. Srilakshmi, Assistant Professor**, who has guided us a lot and encouraged us in every step of project work with her valuable moral support and guidance throughout the project helped us to a great extent.

We wish to express our sincere thanks to **Dr. K. Madhavi, Head** of the Department of CSE for her valuable guidance given throughout. We feel elated to thank **Prof. J. Praveen,** Principal for providing facilities to complete the dissertation in completion of our project and throughout my course.

We would like to thank our faculty whose constructive ideas during reviews helped us in shaping the project better. We would like to thank our parents for inspiring us all the way for arranging all the facilities and resources needed for our project. Not to forget, our non-teaching staff, and friends who had directly or indirectly helped and supported us in completing our project in time.

|  |  |
| --- | --- |
| **Y. BHARATH** | **15241A05H9** |
| **G. HARI KRISHNA** | **15241A05D2** |
| **UZAIR ADEEL** | **15241A05F0** |
| **J. REVANTH** | **15241A05D5** |
| **K. ROHITH KUMAR** | **16245A0529** |

# DECLARATION

We hereby declare that the industrial major project entitled **“Crime Analysis and Prediction”** is the work done during the period from **10th Dec 2018 to 10 April 2019** and is submitted in the partial fulfilment of the requirements forthe award of degree of Bachelor of Technology in Computer Science and Engineering from Gokaraju Rangaraju Institute of Engineering and Technology (Autonomous under Jawaharlal Nehru Technology University, Hyderabad).The results embodied in this project have not been submitted to any other university or Institution for the award of any degree or diploma.

|  |  |
| --- | --- |
| **Y. BHARATH** | **15241A05H9** |
| **G. HARI KRISHNA** | **15241A05D2** |
| **UZAIR ADEEL** | **15241A05F0** |
| **J. REVANTH** | **15241A05D5** |
| **K. ROHITH KUMAR** | **16245A0529** |

# ABSTRACT

Crime analysis and prevention is a systematic approach for identifying and analysing patterns and trends in crime. Our system can predict regions which have high probability for crime occurrence and can visualize crime prone areas. With the increasing advent of computerized systems, crime data analysts can help the Law enforcement officers to speed up the process of solving crimes. Using the concept of data mining we can extract previously unknown, useful information from an unstructured data. Here we have an approach between computer science and criminal justice to develop a data mining procedure that can help solve crimes faster.

# INDEX

[CERTIFICATE i](#_Toc5295212)

[ACKNOWLEDGEMENT ii](#_Toc5295213)

[DECLARATION iii](#_Toc5295214)

[ABSTRACT iv](#_Toc5295215)

[INDEX v](#_Toc5295216)

[1. INTRODUCTION 1](#_Toc5295217)

[1.1 Introduction to Data Mining 1](#_Toc5295218)

[1.1.1 Data Mining Process – Knowledge Discovery in Databases 2](#_Toc5295219)

[1.1.2 Data Mining - Tasks 3](#_Toc5295220)

[1.2 Existing System 3](#_Toc5295221)

[1.3 Proposed System 3](#_Toc5295222)

[2. SYSTEM-ANALYSIS 5](#_Toc5295223)

[2.1 System Requirements: 5](#_Toc5295224)

[2.1.1 Software Requirements: 5](#_Toc5295225)

[2.1.2 Hardware Requirements: 8](#_Toc5295226)

[2.2 Process System: 8](#_Toc5295227)

[3. TECHNOLOGY 10](#_Toc5295228)

[3.1 APIs and Software involved 10](#_Toc5295229)

[3.1.1 Flask Framework 10](#_Toc5295230)

[3.1.2 Lasso Regression Model 11](#_Toc5295231)

[3.1.3 SQLite3 14](#_Toc5295232)

[3.2 Python Libraries: 16](#_Toc5295233)

[3.2.1 NumPy: 16](#_Toc5295234)

[3.2.2 Pandas: 17](#_Toc5295235)

[3.2.3 Matplotlib: 18](#_Toc5295236)

[3.2.4 Seaborn: 19](#_Toc5295237)

[4. ALGORITHM DESCRIPTION 21](#_Toc5295238)

[4.1 Back End: files and their functionality 21](#_Toc5295239)

[4.1.1 Backend: Analysis of two types 22](#_Toc5295240)

[4.1.2 Backend: Process 23](#_Toc5295241)

[4.2 Front End: 24](#_Toc5295242)

[5. SYSTEM DESIGN 25](#_Toc5295243)

[5.1 UML Diagrams 25](#_Toc5295244)

[5.1.1 Use Case Diagram 27](#_Toc5295245)

[5.1.2 Class Diagram 28](#_Toc5295246)

[5.1.3 Activity Diagram 30](#_Toc5295247)

[6. IMPLEMENTATION 31](#_Toc5295248)

[6.1 Code for loading dataset and Importing modules: 31](#_Toc5295249)

[6.1.1 Import modules code 31](#_Toc5295250)

[6.2 Dimension Reduction: 33](#_Toc5295251)

[6.3 Lasso Regression for Prediction 34](#_Toc5295252)

[6.4 Perform Prediction Analysis and Visualization: 36](#_Toc5295253)

[7. SYSTEM TESTING 37](#_Toc5295254)

[7.1 Software testing 37](#_Toc5295255)

[7.2 TYPES OF TESTING 39](#_Toc5295256)

[7.2.1 Black Box Testing 39](#_Toc5295257)

[7.2.2 White Box Testing 40](#_Toc5295258)

[7.3 TESTING PLAN 40](#_Toc5295259)

[7.3.1 Test Data 40](#_Toc5295260)

[7.3.3 Test Report 41](#_Toc5295261)

[7.3.4 Error Report 41](#_Toc5295262)

[8. SCREENSHOTS 46](#_Toc5295263)

[9. CONCLUSION 48](#_Toc5295264)

[9.1 Conclusion: 48](#_Toc5295265)

[9.2 Future Scope: 49](#_Toc5295266)

[10. BIBLIOGRAPHY 50](#_Toc5295267)

[10.1 References 50](#_Toc5295268)

# 1. INTRODUCTION

## 1.1 Introduction to Data Mining

Data mining is the process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems. Data mining is an interdisciplinary subfield of computer science with an overall goal to extract information (with intelligent methods) from a data set and transform the information into a comprehensible structure for further use. Data mining is the analysis step of the "knowledge discovery in databases" process, or KDD. Aside from the raw analysis step, it also involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, postprocessing of discovered structures, visualization, and online updating.

The information or knowledge extracted so can be used for any of the following applications −

* Market Analysis
* Fraud Detection
* Customer Retention
* Production Control
* Science Exploration

Data mining is highly useful in the following domains −

* Market Analysis and Management
* Corporate Analysis & Risk Management
* Fraud Detection

### Data Mining Process – Knowledge Discovery in Databases

The knowledge discovery in databases (KDD) process is commonly defined with the stages:

1. **Selection-**

Where data relevant to the analysis task are retrieved from the database

1. **Pre-processing-**

Data pre-processing 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 behaviours or trends, and is likely to contain many errors.

1. **Transformation-**

Where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance.

1. **Data mining-**

An essential process where intelligent methods are applied in order to extract data patterns.

1. **Interpretation/evaluation-**

Pattern evaluation to identify the truly interesting patterns representing knowledge based on some interestingness measures; Knowledge presentation where visualization and knowledge representation techniques are used to present the mined knowledge to the user.

### 1.1.2 Data Mining - Tasks

**Data mining involves six common classes of tasks:**

* Anomaly detection (outlier/change/deviation detection) – The identification of unusual data records, that might be interesting or data errors that require further investigation.
* Association rule learning (dependency modelling) – Searches for relationships between variables. For example, a supermarket might gather data on customer purchasing habits. Using association rule learning, the supermarket can determine which products are frequently bought together and use this information for marketing purposes. This is sometimes referred to as market basket analysis.
* Clustering – is the task of discovering groups and structures in the data that are in some way or another "similar", without using known structures in the data.
* Classification – is the task of generalizing known structure to apply to new data. For example, an e-mail program might attempt to classify an e-mail as "legitimate" or as "spam".
* Regression – attempts to find a function which models the data with the least error that is, for estimating the relationships among data or datasets.
* Summarization – providing a more compact representation of the data set, including visualization and report generation.

## 1.2 Existing System

Day by day the crime rate is increasing considerably. Crime cannot be predicted since it is neither systematic nor random. Also, the modern technologies and hi-tech methods help criminals in achieving their misdeeds.

There are analyses made on different domains like Market analysis and prediction, Fraud Detection etc.

However, the existing applications do not showcase a good accuracy.

## 1.3 Proposed System

Since the crime rate is increasing day by day, our goal is to propose a software for predicting when and where a particular crime will take place. Our Analysis would contain analysis of crime rates of Mexico, which includes Descriptive Analysis and Predictive Analysis.

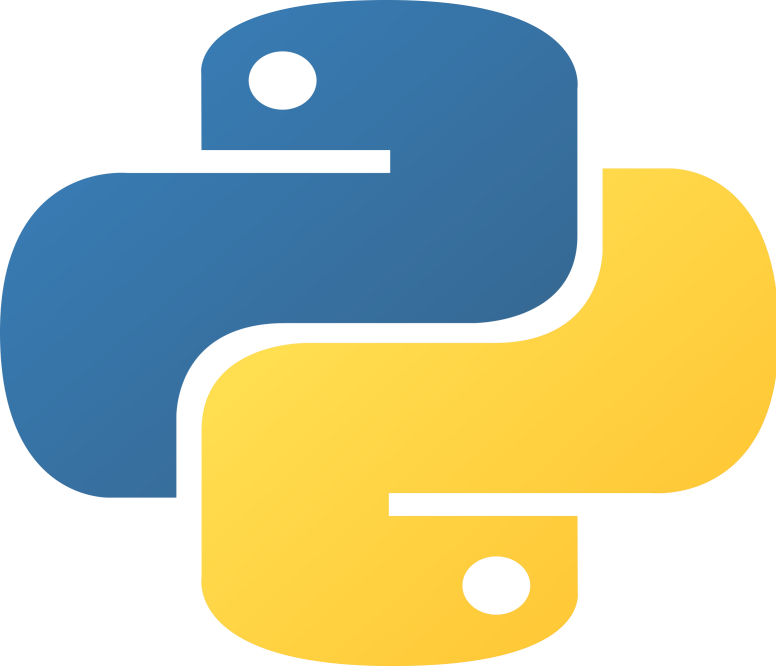
# 2. SYSTEM-ANALYSIS

## 2.1 System Requirements:

**2.1.1 Software Requirements:**

**a) Python:**

Python is a deciphered language Guido van Rossum, Python has a diagram hypothesis that complements code decipherability, and a sentence structure that empowers programming architects to express thoughts in less lines of code noticeably using imperative whitespace. It gives builds up that engage little immense. Incorporates a kind modified organization. Reinforces different perfect models, masterminded, essential, useful and, and a huge and exhaustive.



#### Fig 2.1 **Python**

Python translators are accessible some working structures, references of, is an software a get-together change appear, as about the majority of its assortment utilize. Python is directed Programming Establishment execution was started by Wakened

and to the ABC dialect arranged for extraordinary case managing the Amoeba working structure. Van Rossum remains Python's main maker. His methodology and support for Unicode. With this discharge, the switch framework wound up being all the more clear and assembling kept up. An amazing piece of its authentic been backported to the 2.7.x variety strategy, and arrivals of Python 3 automates the comprehension of Python 2 code to Python 3.

Python 2. complete - date () was at first set at2015, by then put off to 2020 out of stress that an enormous gathering of code couldn't without a lot of an extend to Python 3. Python 3.6 had changes with regard to UTF-8 and Python 3.7.0b1 includes another "UTF-8 Mode”. Python is a multi-viewpoint programming vernacular. Question coordinated programming and dealt with creating PC programs are completely kept up, and incalculable highlights bolster practical programming and point orchestrated programming (checking by metaprogramming and metaobjects (engage methodology)).

Different assorted immaculate models are kept up through developments, including plan by and reason uses written work, and a mix -recognizing city labourer for organization. Moreover, incorporates assurance (late definitive), ties procedure and in the midst of Python’s outline offers some help for feasible programming in the Lisp convention. It has channel(), design(), diminish() limits; list much appreciated, word references, and sets; and generator clarifications. The dialect's centre rationality is abridged in the archive The Zen of Python, which incorporates maxims, for example,

1. Delightful appalling
2. Unequivocal is to understood
3. Straightforward is to complex
4. is to be entangled
5. Decipherability tallies

Other of having most of its value joined with its centre, was planned to be exceedingly littler segregation understood as a strategy for adding a little focus lingo with a generous and successfully arbiter began from his mistake with, which grasped the. giving decision in system, rationality does not accept rich linguistic, (for example, that of Perl) for an easier, very less-jumbled language. Martelit: "To portray something as 'cunning' isn't viewed as a compliment in the Python culture." does not accept the Perl "there is in excess of one approach to do it” for "there ought to be one— and ideally just a single evident approach to do it".

Specialists attempt avoids awkward improvement and reject fewer fringe augments. At modules written in tongues, for instance, C, or use PyPy, a just under the wire.

A crucial goal of Python's creators is keeping it agreeable to use. This is reflected in the lingo's name—a tribute to the English comic show gather Monty Python— and in sporadically direct soundness and emphasis on seriousness. Alternately, code that is difficult to understand or scrutinizes like an unforgiving tongue is called unpythonic.

**b) Flask Framework**

**Flask** is a microweb framewor[k](https://en.wikipedia.org/wiki/Web_framework) written in [Python.](https://en.wikipedia.org/wiki/Python_(programming_language)) It is classified as a [microframework](https://en.wikipedia.org/wiki/Microframework) because it does not require particular tools or libraries.[[3]](https://en.wikipedia.org/wiki/Flask_(web_framework)#cite_note-3) It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more regularly than the core Flask program.[[4]](https://en.wikipedia.org/wiki/Flask_(web_framework)#cite_note-4) Flask is commonly used with MongoDB, which gives it more control over databases and history.

1. **SQLite3 querying tool for the data base**

SQLite is an in-process library that implements a self-contained, serverless, zero configuration, transactional SQL database engine. It is a database, which is zero configured, which means like other databases you do not need to configure it in your system. SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

1. **Ubuntu or LINUX based operating system**

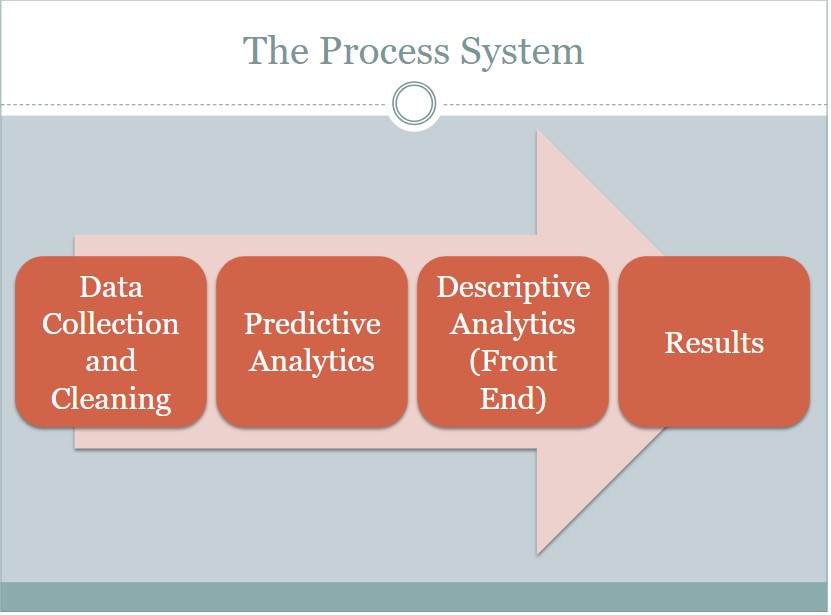
Since we are using .csv files to convert into data frames for processing, a Linux based operating system is required to overcome the problem that occurs with UTF-8 Encoding.

### 2.1.2 Hardware Requirements:

RAM: 4GB and Higher

Processor: Intel i3 and above

## 2.2 Process Flow:



#### **Fig 2.2** Process Flow

# 3. TECHNOLOGY

## 3.1 APIs and Software involved

### 3.1.1 Flask Framework

**What is a Web Application Framework?**

Web Application Framework or simply Web Framework represents a collection of libraries and modules that enables a web application developer to write applications without having to bother about low-level details such as protocols, thread management etc.

**Flask Framework**

Flask is a web application framework written in Python. It is developed by **Armin Ronacher**, who leads an international group of Python enthusiasts named Pocco. Flask is based on the Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

* **WSGI**

Web Server Gateway Interface (WSGI) has been adopted as a standard for Python web application development. WSGI is a specification for a universal interface between the web server and the web applications.

* **Werkzeug**

It is a WSGI toolkit, which implements requests, response objects, and other utility functions. This enables building a web framework on top of it. The Flask framework uses Werkzeug as one of its bases.

* **jinja2**

jinja2 is a popular templating engine for Python. A web templating system combines a template with a certain data source to render dynamic web pages. Flask is often referred to as a micro framework. It aims to keep the core of an application simple yet extensible. Flask does not have built-in abstraction layer for database handling, nor does it have form a validation support. Instead, Flask supports the extensions to add such functionality to the application.

### 3.1.2 Lasso Regression Model

In statistics and machine learning, lasso (least absolute shrinkage and selection operator; also Lasso or LASSO) is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces. It was originally introduced in geophysics literature in 1986, and later independently rediscovered and popularized in 1996 by Robert Tibshirani, who coined the term and provided further insights into the observed performance.

Lasso was originally formulated for least squares models and this simple case reveals a substantial amount about the behaviour of the estimator, including its relationship to ridge regression and best subset selection and the connections between lasso coefficient estimates and so-called soft thresholding. It also reveals that (like standard linear regression) the coefficient estimates need not be unique if covariates are collinear.

Lasso regression is one of the regularization methods that creates parsimonious models in the presence of large number of features, where large means either of the below two things:

1. Large enough to enhance the tendency of the model to over-fit. Minimum ten variables can cause overfitting.

1. Large enough to cause computational challenges. This situation can arise in case of millions or billions of features.

Lasso regression performs L1 regularization that is it adds the penalty equivalent to the absolute value of the magnitude of the coefficients. Here the minimization objective is as followed.

Minimization objective = LS Obj + λ (sum of absolute value of coefficients)

Where LS Obj stands for Least Squares Objective which is nothing but the linear regression objective without regularization and λ is the turning factor that controls the amount of regularization. The bias will increase with the increasing value of λ and the variance will decrease as the amount of shrinkage (λ) increases.

**When Lasso regression was developed and what is its purpose?**

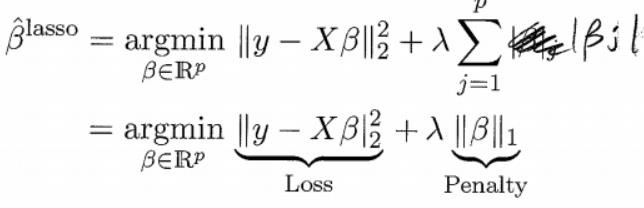
The lasso regression model was developed in 1989. It is basically an alternative to the classic least squares estimate to avoid many of the problems with overfitting when we have a large number of independent variables.

**What large coefficient signifies?**

By using large coefficient, we are putting a huge emphasis on the particular feature that it can be a good predictor of the outcome. And when it is too large, the algorithm starts modelling intricate relations to calculate the output & ends up overfitting to the particular data. Lasso regression adds a factor of the sum of the absolute value of the coefficients the optimization objective.

Now let us understand lasso regression formula with a working example:

The lasso regression estimate is defined as



Here the turning factor λ controls the strength of penalty, that is

When λ = 0: We get same coefficients as simple linear regression

When λ = ∞: All coefficients are zero

When 0 < λ < ∞: We get coefficients between 0 and that of simple linear regression

So, when λ is in between the two extremes, we are balancing the below two ideas.

1. Fitting a linear model of y on X
2. Shrinking the coefficients

But the nature of L1 regularization penalty causes some coefficients to be shrunken to zero. Hence, unlike ridge regression, lasso regression is able to perform variable selection in the liner model. So as the value of λ increases, more coefficients will be set to value zero (provided fewer variables are selected) and so among the nonzero coefficients, more shrinkage will be employed. The below working example will explain it well.

### 3.1.3 SQLite3

SQLite is an in-process library that implements a self-contained, serverless, zero configuration, transactional SQL database engine. It is a database, which is zero configured, which means like other databases you do not need to configure it in your system.

SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. SQLite accesses its storage files directly.

**Why SQLite?**

* SQLite does not require a separate server process or system to operate (serverless).
* SQLite comes with zero-configuration, which means no setup or administration needed.
* A complete SQLite database is stored in a single cross-platform disk file.
* SQLite is very small and light weight, less than 400KiB fully configured or less than 250KiB with optional features omitted.
* SQLite is self-contained, which means no external dependencies.
* SQLite transactions are fully ACID-compliant, allowing safe access from multiple processes or threads.
* SQLite supports most of the query language features found in SQL92 (SQL2) standard.
* SQLite is written in ANSI-C and provides simple and easy-to-use API.
* SQLite is available on UNIX (Linux, Mac OS-X, Android, iOS) and Windows (Win32, WinCE, WinRT).

**Install SQLite on Windows**

* Step 1 − Go to [SQLite download page,](https://www.sqlite.org/download.html) and download precompiled binaries from Windows section.
* Step 2 − Download sqlite-shell-win32-\*.zip and sqlite-dll-win32-\*.zip zipped files.
* Step 3 − Create a folder C:\>sqlite and unzip above two zipped files in this folder, which will give you sqlite3.def, sqlite3.dll and sqlite3.exe files.
* Step 4 − Add C:\>sqlite in your PATH environment variable and finally go to the command prompt and issue sqlite3 command, which should display the following result.

|  |
| --- |
| C:\>sqlite3  SQLite version 3.7.15.2 2013-01-09 11:53:05  Enter ".help" for instructions  Enter SQL statements terminated with a ";"  Sqlite> |

#### Fig 3.1: Sqlite3 prompt

## 3.2 Python Libraries:

### 3.2.1 NumPy:

NumPy is a library for the Python programming vernacular, including support for colossal, multi-dimensional bunches and cross sections, nearby a significant get-together of unusual state mathematical abilities to chip away at these shows. The forerunner of NumPy, Numeric, was at first made by Jim Hugunin with duties from a couple of various fashioners. Using NumPy in Python gives value identical to MATLAB since they are both interpreted, and they both empower the customer to make speedy ventures as long as most undertakings wear down groups or networks as opposed to scalars. In examination, MATLAB boasts a far-reaching number of additional instrument stash, exceptionally Simulink, however NumPy is naturally planned with Python, a more present day and complete the process of programming lingo. Also, correlative Python groups are available; SciPy is a library that incorporates more MATLAB-like value and Matplotlib is a plotting pack that gives MATLAB-like plotting convenience.

Inside, both MATLAB and NumPy rely upon BLAS and LAPACK for compelling straight factor-based math counts. Python ties of the extensively used PC vision library OpenCV utilize NumPy displays to store and work on data. Since pictures with various channels are essentially addressed as three-dimensional shows, requesting, cutting or covering with various groups are especially powerful ways to deal with get to specific pixels of a photo. The NumPy display as general data structure in OpenCV for pictures, evacuated component centres, channel bits and various more vastly streamlines the programming work process and exploring.

### 3.2.2 Pandas:

In PC programming, pandas is an item library made for the Python programming vernacular for data control and examination. It offers data structures and exercises for controlling numerical tables and time course of action. It is free programming released under the three-condition BSD allow.

The name is gotten from the articulation "load up data", an econometrics term for instructive files that fuse both time-course of action and cross-sectional data. Pandas is a NumFocus up held project. This will help ensure the achievement of progression of pandas as a world-class open-source project, and makes it possible to provide for the wander

Library Features:

1. Data Frame question for information control with incorporated ordering.
2. Tools for perusing and composing information between in-memory information structures and distinctive record groups.
3. Data arrangement and incorporated treatment of missing information.
4. Reshaping and rotating of informational indexes.
5. Label-based cutting, extravagant ordering, and sub setting of substantial informational indexes.
6. Data structure segment inclusion and cancellation.
7. Group by motor permitting split-apply-join tasks on informational indexes.
8. Data set consolidating and joining.
9. Hierarchical pivot ordering
10. Hierarchical pivot ordering to work with high-dimensional information in a lower-dimensional information structure.
11. Time arrangement usefulness: Date run age and recurrence transformation, moving window measurements, moving window direct relapses, date moving and slacking.

### 3.2.3 Matplotlib:

Matplotlib is a plotting library for the Python programming dialect and its numerical science expansion NumPy. It gives a protest arranged API to installing plots into applications utilizing broadly useful GUI toolboxes’ like Tkintr, wxPython, Qt, or GTK+. There is additionally a procedural "pylab" interface in view of a state machine (like OpenGL), intended to nearly look like that of MATLAB, however its utilization is debilitated. SciPy makes utilization of matplotlib. matplotlib was initially composed by John D. Seeker, has a dynamic advancement group, and is conveyed under a BSD-style permit. Michael Droettboom was designated as matplotlib’s lead engineer in the blink of an eye before John Hunter's demise in 2012.

**Examples:**

**i) Line Plot:**

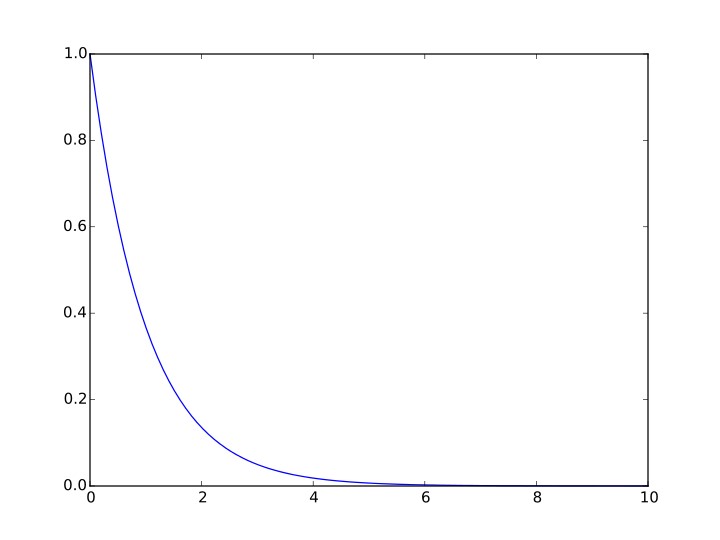
>import matplotlib .pylot as plt >import numpy as np

>a=np.linspace(0,10,100)

>b=np.exp(-a)

>plt.plot(a,b)

>Plt.show()



#### **Fig 2.2** Line Plot

**ii) Histogram:**

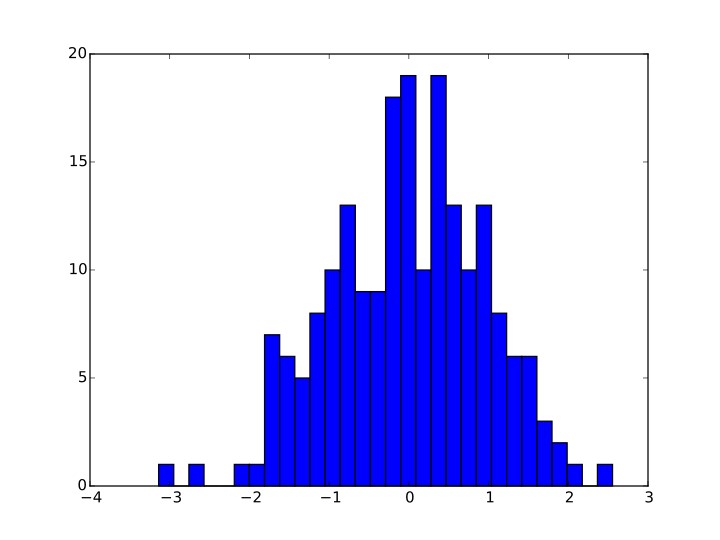
>import matplotlib.pyplot as plt

>from NumPy Random import normal, and

>x=normal(size=200)

>plt.hit(x,bins=30)

>plt.show()



#### **Fig 2.3** Histogram plot

### 3.2.4 Seaborn:

Seaborn is a library for making statistical graphics in Python. It is built on top of [matplotlib](https://matplotlib.org/) and closely integrated with [pandas](https://pandas.pydata.org/) data structures.

Here is some of the functionality that seaborn offers:

* A dataset-oriented API for examining [relationships](https://seaborn.pydata.org/examples/scatter_bubbles.html#scatter-bubbles) between [multiple variables](https://seaborn.pydata.org/examples/faceted_lineplot.html#faceted-lineplot)
* Specialized support for using categorical variables to show [observations](https://seaborn.pydata.org/examples/jitter_stripplot.html#jitter-stripplot) or [aggregate statistics](https://seaborn.pydata.org/examples/pointplot_anova.html#pointplot-anova)
* Options for visualizing [univariate](https://seaborn.pydata.org/examples/distplot_options.html#distplot-options) or [bivariate](https://seaborn.pydata.org/examples/joint_kde.html#joint-kde) distributions and for [comparing](https://seaborn.pydata.org/examples/horizontal_boxplot.html#horizontal-boxplot) them between subsets of data
* Automatic estimation and plotting of [linear regression](https://seaborn.pydata.org/examples/anscombes_quartet.html#anscombes-quartet) models for different kinds [dependent](https://seaborn.pydata.org/examples/logistic_regression.html#logistic-regression) variables
* Convenient views onto the overall [structure](https://seaborn.pydata.org/examples/scatterplot_matrix.html#scatterplot-matrix) of complex datasets
* High-level abstractions for structuring [multi-plot grids](https://seaborn.pydata.org/examples/faceted_histogram.html#faceted-histogram) that let you easily build [complex](https://seaborn.pydata.org/examples/pair_grid_with_kde.html#pair-grid-with-kde) visualizations
* Concise control over matplotlib figure styling with several [built-in themes](https://seaborn.pydata.org/tutorial/aesthetics.html#aesthetics-tutorial)
* Tools for choosing [colour palettes](https://seaborn.pydata.org/tutorial/color_palettes.html#palette-tutorial) that faithfully reveal patterns in your data

Seaborn aims to make visualization a central part of exploring and understanding data. Its dataset-oriented plotting functions operate on data frames and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots.

# 4. ALGORITHM DESCRIPTION

## 4.1 Back End: files and their functionality

Here is where the python scripts to create the data-frame, clean the data-frame, run a forecast model using the data-frame and retrieve the forecast results are located.

1. **crime\_process.py**

Inputs mini\_db/raw\_csv/crime.csv and reshape it.

Create mini\_db/process\_csv/crime.cvs.

This is the file that process the predicted columns (crimes).

1. **csv\_process.py**

Inputs mini\_db/raw\_csv/n.csv and reshape them.

Where n are the different tables of the predictors (justice\_system, education, health, etc.). Create mini\_db/process\_csv/n.cvs.

This is the file that process the predictors (covariates) columns.

1. **dataframe\_lasso.py**

Inputs mini\_db.db creates a connection between sqlite and python3 and makes a pandas dataframe. Constructs changes and lags of the predicted columns and predictors. Drop NaN and entites where there is not complete information. Restricts the dataframe for the years 2008-2011 (where there is complete information). Outputs df\_limited.csv and lists: list\_dep (dependent) and list\_lag (covariates).

1. **lasso\_model.py**

Inputs the outputs of dataframe\_lasso.

Construct function frontend that inputs name of the variable to predict (string) and year. Outputs: actual observations, predicted observations, coefficients of the predictors, mean square error (mse), corr(pred,actual) and r-square of the prediction.

Construct a class where the estimations of the predictive model (lasso) are done.

### 4.1.1 Backend: Analysis of two types

Two types of analyses are shown for the result which are,

**Descriptive Analysis**

This is the analysis where the actual integrated and cleaned data is represented by Visualizations such as bar graphs etc using heatmaps of matplotlib library in python.

**Predictive Analysis**

This is the analysis where a prediction of when and where a crime in a year would take place by training the model with a data set of crime rates in Mexico. Lasso Regression model is used for Prediction.

### 4.1.2 Backend: Process

**1. Data Integration and Cleaning**

The information is gathered using a dataset having crime rates of Mexico. Using seven different tables to standardize and format the dataset. The data is in a .csv file which is imported in the backend to clean the data to get rid of the outliers, noisy and inconsistent data.

* 1. **Standardize and merging**

Analysing independent and dependent variables and integrates them applying required operations.

* 1. **Perform Analysis and forecasting Techniques**

Predictive and descriptive analyses are performed using Lasso Regression of Statistics,

**LASSO** stands for **Least Absolute Shrinkage and Selection Operator**. Lasso regression is one of the regularization methods that creates parsimonious models in the presence of large number of features, where large means either of the below two things:

1. Large enough to enhance the tendency of the model to over-fit. Minimum ten variables can cause overfitting.

2. Large enough to cause computational challenges. This situation can arise in case of millions or billions of features.

Lasso regression performs L1 regularization that is it adds the penalty equivalent to the absolute value of the magnitude of the coefficients. Here the minimization objective is as followed.

Minimization objective = LS Obj + λ (sum of absolute value of coefficients) Where LS Obj stands for Least Squares Objective which is nothing but the linear regression objective without regularization and λ is the turning factor that controls the amount of regularization. The bias will increase with the increasing value of λ and the variance will decrease as the amount of shrinkage (λ) increases.

## 4.2 Front End:

There are three main files in the frontend folder:

1. Templates folder: a folder for all the Jinja/html templates for Flask.
2. Static: a JavaScript file for charts.
3. flask\_frontend\_crime.py: A python file with the code to run the website.

In the Front end all the templates are written in html along with CSS which are to be displayed on to the screen when they are run using flask framework that integrates front end and backend and a python code is written in a file that supports visualisation functions and databases to obtain the results.

# 5. SYSTEM DESIGN

## 5.1 UML Diagrams

Unified Modelling Language is a tool that helps a designer to present his ideas about the project to his client and his developer. Modelling plays a crucial role in designing a software. A poorly designed model can lead to a poorly developed software.

A UML system has using five different views that help in describing systems from different perspectives. Each view has a set of diagrams that and components that represent the real time objects.

* **User Model View:** 
  1. It models the user behaviour in a system context.
  2. All the diagrams are drawn keeping in mind the user’s response and reaction towards a system.
* **Structural Model View** 
  1. This view consists of class diagram and object diagram which is used to model the static structures.
  2. It uses objects, attributes, operations and relationships.
* **Behavioural Model View** 
  1. It mainly consists of the sequence diagram, collaboration diagram, state chart diagram and activity diagram. They mainly represent flow of actions between different objects involved in the system
  2. They are used to visualize various dynamic aspects of the system architecture.

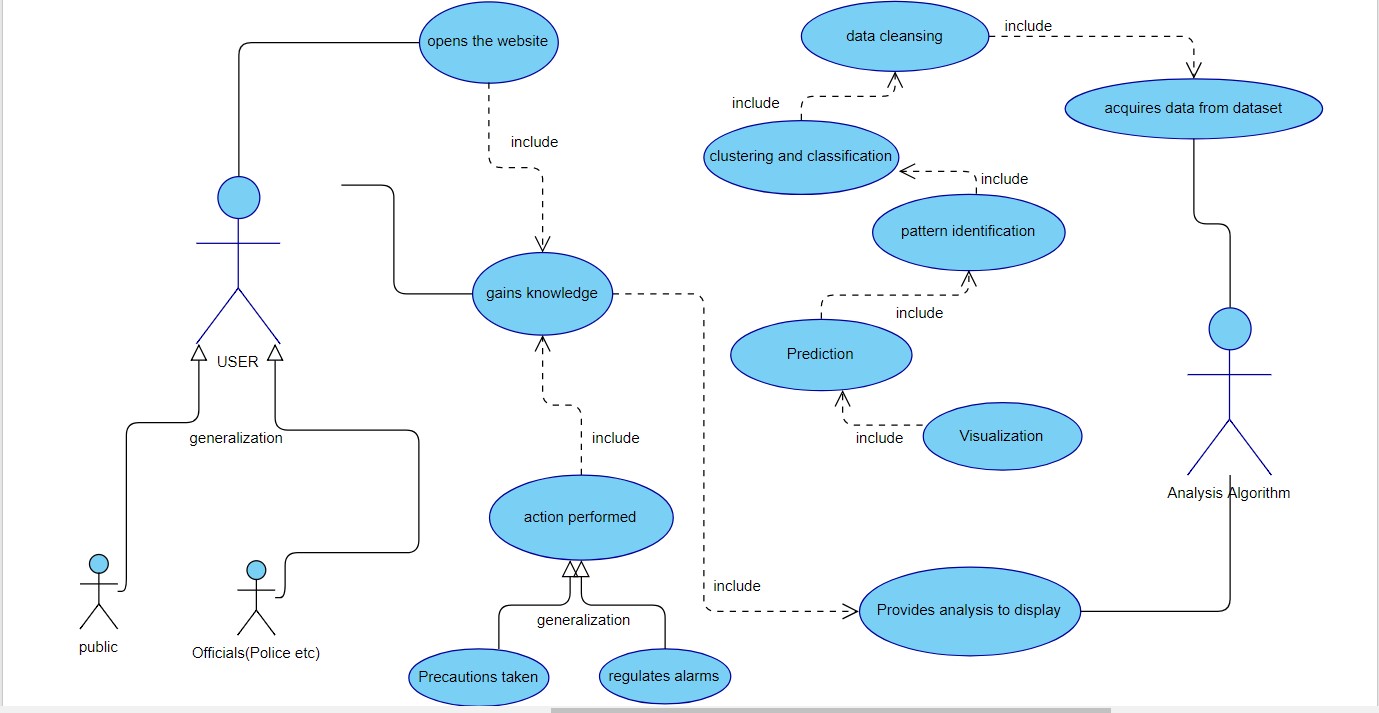
* **Implementation Model View** 
  1. This view consists of component diagrams and deployment diagrams. This view models the static software modules for an organization.
  2. This usually contains the data files, documentation, the executables and source code.

### 5.1.1 Use Case Diagram

The basic representation for the interaction of the user with the system is represented using the use case diagram. It involves the relationship between the user and various use cases with the actors being involved. There are different kinds of relationships that are involved between the use cases and the actors. They include:

1. Association relationship
2. Generalization
3. Dependency
4. Realizations
5. Transitions

The following represents the use case diagram of the proposed system:



#### **Fig 5.1:** Use Case Diagram for Developed Model

### 5.1.2 Class Diagram

They are static representation of an application. Only the class diagrams have the capability to be directly mapped with the OOP Languages because in OOPs everything is model in the form of classes and objects. Because of this reason these diagrams are used widely at the time of construction. This is one of the most popularly used UML diagram in the designer community. A class diagram plays an essential role in forward and reverse engineering.

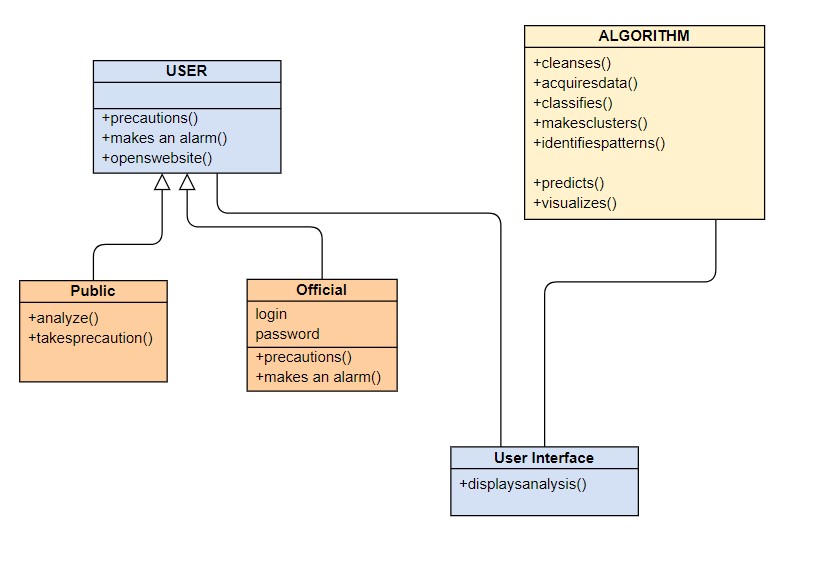
1. It acts as a base for the component and deployment diagrams.
2. It mainly describes and defines the basic responsibilities of a system’s application.
3. It implements the analysis and design view for a static application.

In a class diagram, each object is modelled as a class. Each class consists of section or compartments.

1. Class name
2. Attributes of a class or operations
3. Methods or functions
4. Documentation (optional section)

The following points ought to be recollected while drawing a class diagram:

1. The name of the class diagram must be meaningful to portray the aspect of the framework.
2. Each component and their connections must be distinguished ahead of time.
3. Each class has a responsibility (attributes and methods) that must be identified clearly.
4. Number of properties for each class must be minimum. Since pointless properties will make the diagram convoluted.
5. At whatever point required to depict some part of the diagram use notes Since toward the finish of the diagram it must be justifiable to the designer/coder.

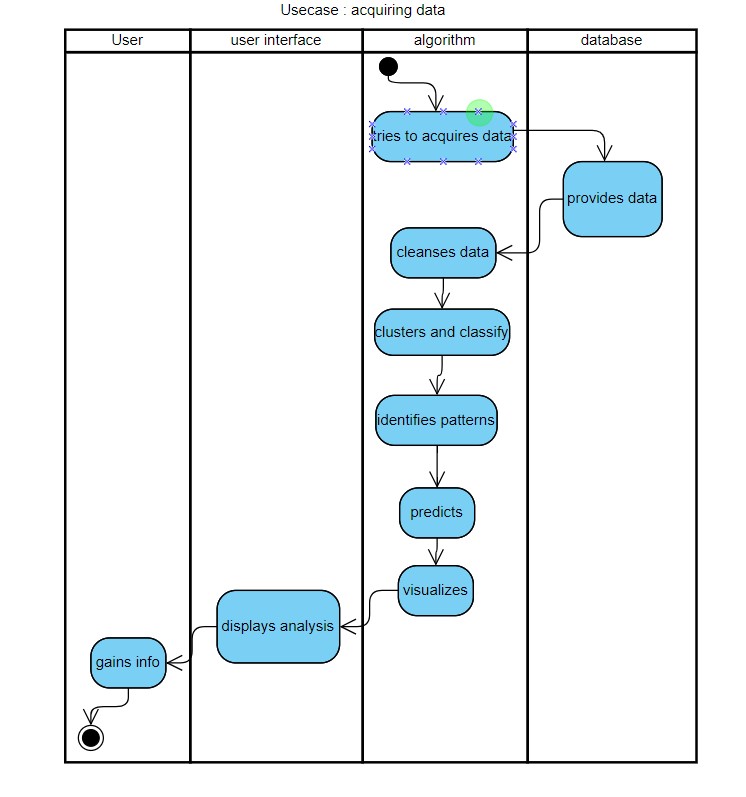


#### **Fig 5.2:** Class Diagram for Developed Model

### 5.1.3 Activity Diagram

The flow from one activity to another activity can be represented in the form of a flow chart which is usually an activity diagram. It forms a backbone for the UML diagrams. It depicts the dynamic aspects for all the objects within the system.

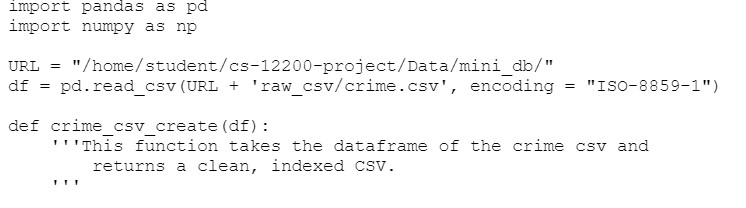
The control flow from one object to another object is drawn which shows the basic operations that are to be performed.



#### **Fig 5.3:** Activity Diagram

# 6. IMPLEMENTATION

## 6.1 Code for loading dataset and Importing modules:

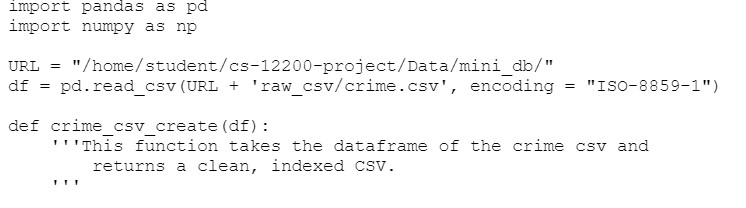


#### **Fig 6.1:** Here we are importing our dataset file (crime.csv).

### 6.1.1 Import modules code

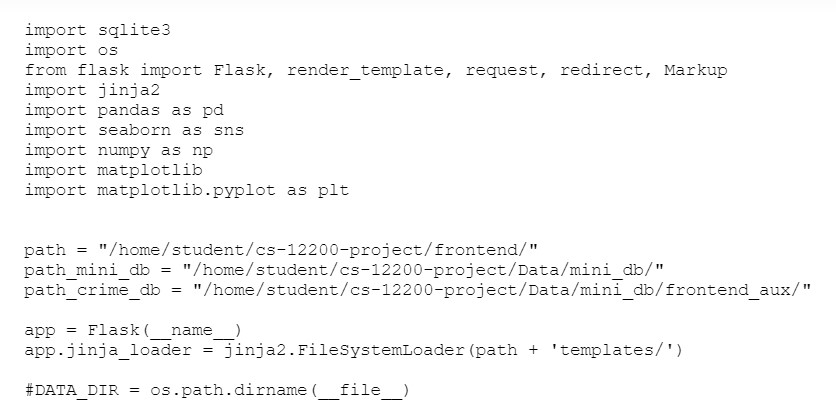
Importing the required modules and libraries to split the dataset, to apply prediction.

**Crime\_process.py:**



#### **Fig 6.2:** Importing modules

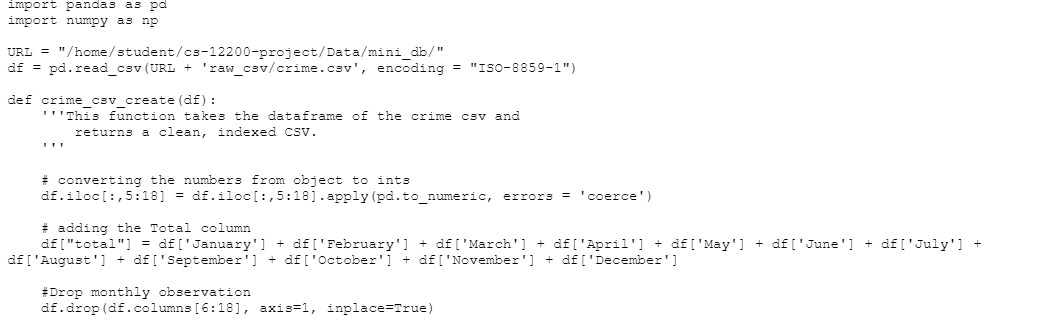
**Front end for visualization- flask\_frontend\_crime.py:**



#### **Fig 6.3:** Visualization module

## 6.2 Dimension Reduction:

The datasets and databases imported are to be cleaned and reduced for finding out the dependent and independent variables or covariates to perform prediction and analysis over the data. The cleaning code is as follows

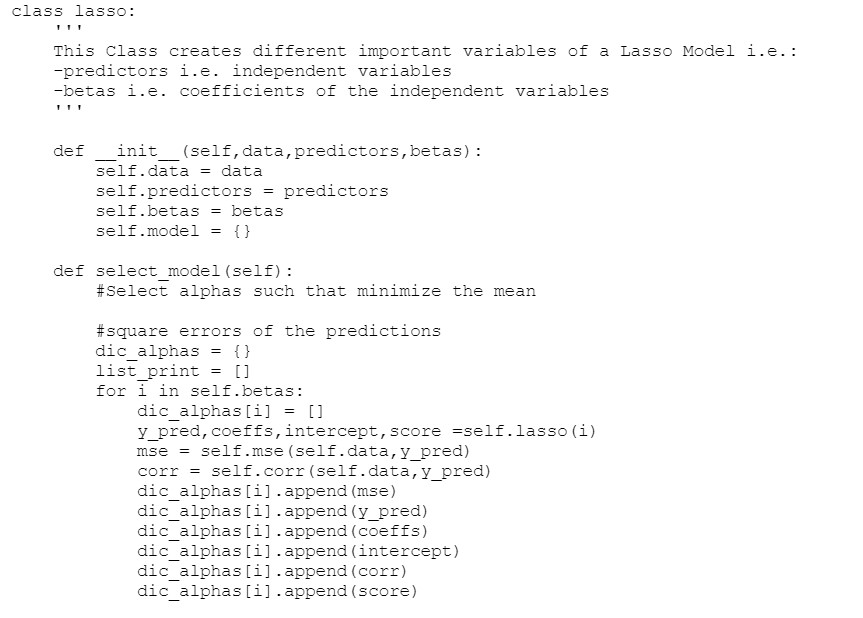


#### **Fig 6.4:** Cleaning module

## 6.3 Lasso Regression for Prediction

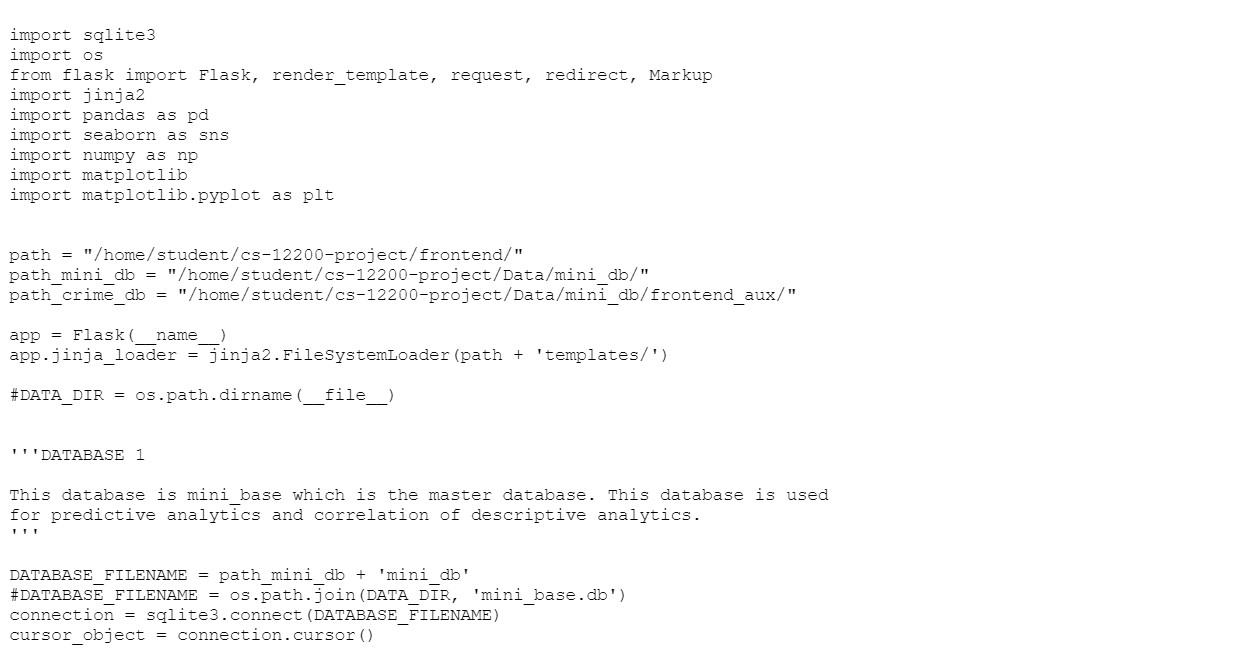
After successful loading datasets, databases and importing the modules we need to perform a regression for prediction.

#### **Fig 6.5:** Running Lasso Algorithm



#### **Fig 6.6:** Class Lasso

## 6.4 Perform Prediction Analysis and Visualization:



#### **Fig 6.7:** Prediction module

# 7. SYSTEM TESTING

## 7.1 Software testing

Software testing is one of the main stages of project development life cycle to provide our cessation utilizer with information about the quality of the application and ours, in our Project we have under gone some stages of testing like unit testing where it’s done in development stage of the project when we are in implementation of the application after the Project is yare we have done manual testing with different Case of all the different modules in the application we have even done browser compatibility testing in different web browsers in market, even we have done Client side validation testing on our application

1. **Unit testing**

The unit testing is done in the stage of implementation of the project only the error is solved in development stage some of the error we come across in development are given below

1. **Manual Testing**

As our Project is academic Leave we can do any automatic testing, so we follow manual testing by endeavour and error methods

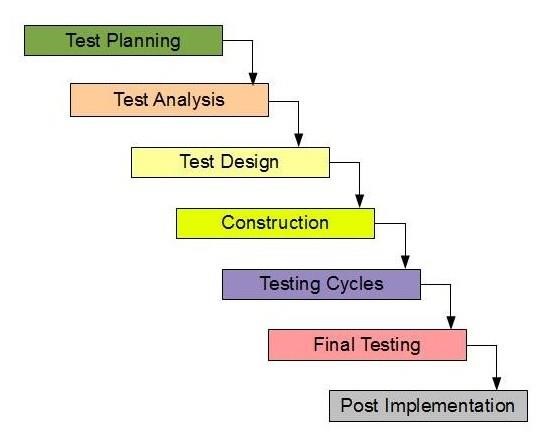
1. **Deployment of System**

Once the project is total yare we will come to deployment of client system in genuinely world as its academic leave we did deployment i our college lab only with all need Software’s with having Windows OS.

1. **Maintenance**

The Maintenance of our Project is one-time process only

|  |  |
| --- | --- |
| Client Needs | Acceptance Testing |
| Requirements | System Testing |
| Design | Integration Testing |
| Code | Unit Testing |



#### **Fig7.1:** Testing phases

Testing is the way toward discovering contrast between the normal conduct indicated by framework models and the watched conduct of the actualized system.

## 7.2 TYPES OF TESTING

There are 2 sorts of testing. They are given beneath:

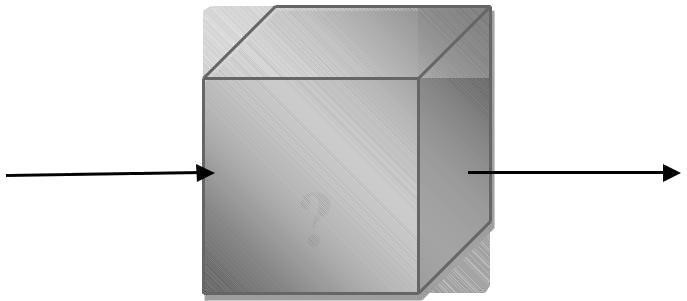
1. Black box or functional testing
2. White box testing or structural testing

### 7.2.1 Black Box Testing

This strategy is used when learning of the predetermined capacity that an item has been intended to perform testing is known. The idea of black box is used for a system whose inside workings are not accessible to examination. In a black box the test thing is a "black", since its rationale is obscure, all that is known in the thing that goes in and what turns out, or the information and the yield.

Black box testing attempts to find errors in the following categories:

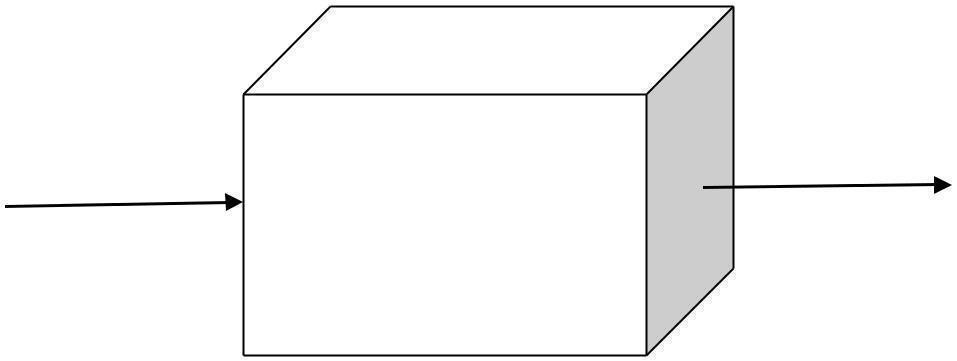
1. Wrong or missing functionalities
2. Errors in data structure
3. Initialization and termination errors
4. Interface errors
5. Execution errors



Input Output

**Fig 7.2:**  Black-box-testing

### 7.2.2 White Box Testing



Input

Output

#### **Fig 7.3:** White Box-testing

White box testing is about the testing that is related to the execution of the program. The plan of basic isn't to practice every one of the sources of info to practice the diverse programming and information structure used as a part of the program. Along these lines, Acceptance testing plans to accomplish test cases that will constrain the scope of various structures.

## 7.3 TESTING PLAN

Testing process starts with a test plan. This plan identifies all the testing related activities that must be performed and specifies the schedules, allocates the resources, and specified guidelines for testing. The final output of the testing phase is the test report and the error report.

### 7.3.1 Test Data

Testing process begins with a test design. This arrangement recognizes all the testing related exercises that must be performed like the timetables, assigning the assets, and determining rules for testing.

**7.3.2 Unit Testing**

Every individual module has been tried against the necessity with some test information.

### 7.3.3 Test Report

The module is working appropriately given the client must enter data. All information section frames have tested with indicated test cases and all information passage shapes are working properly.

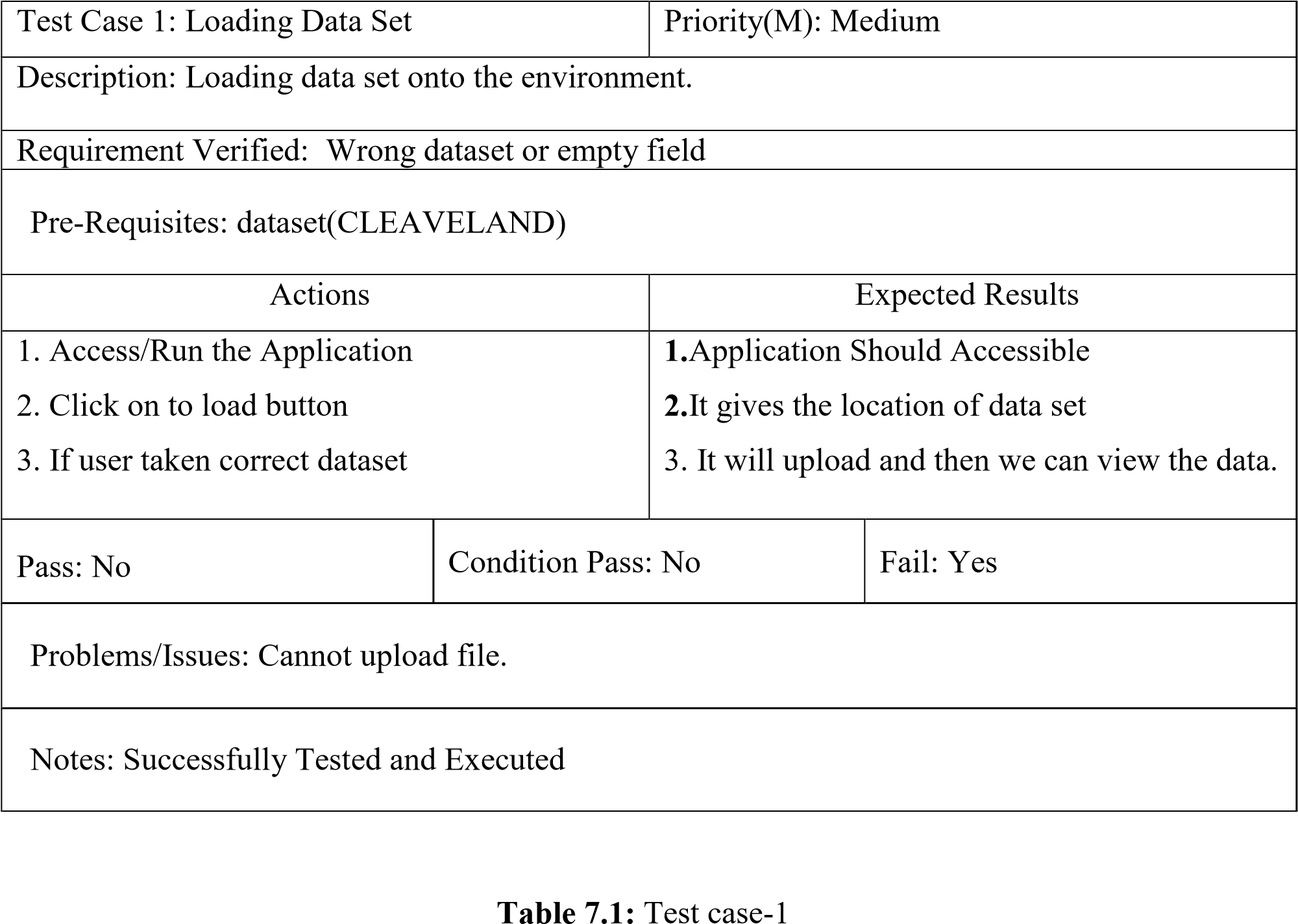
### 7.3.4 Error Report

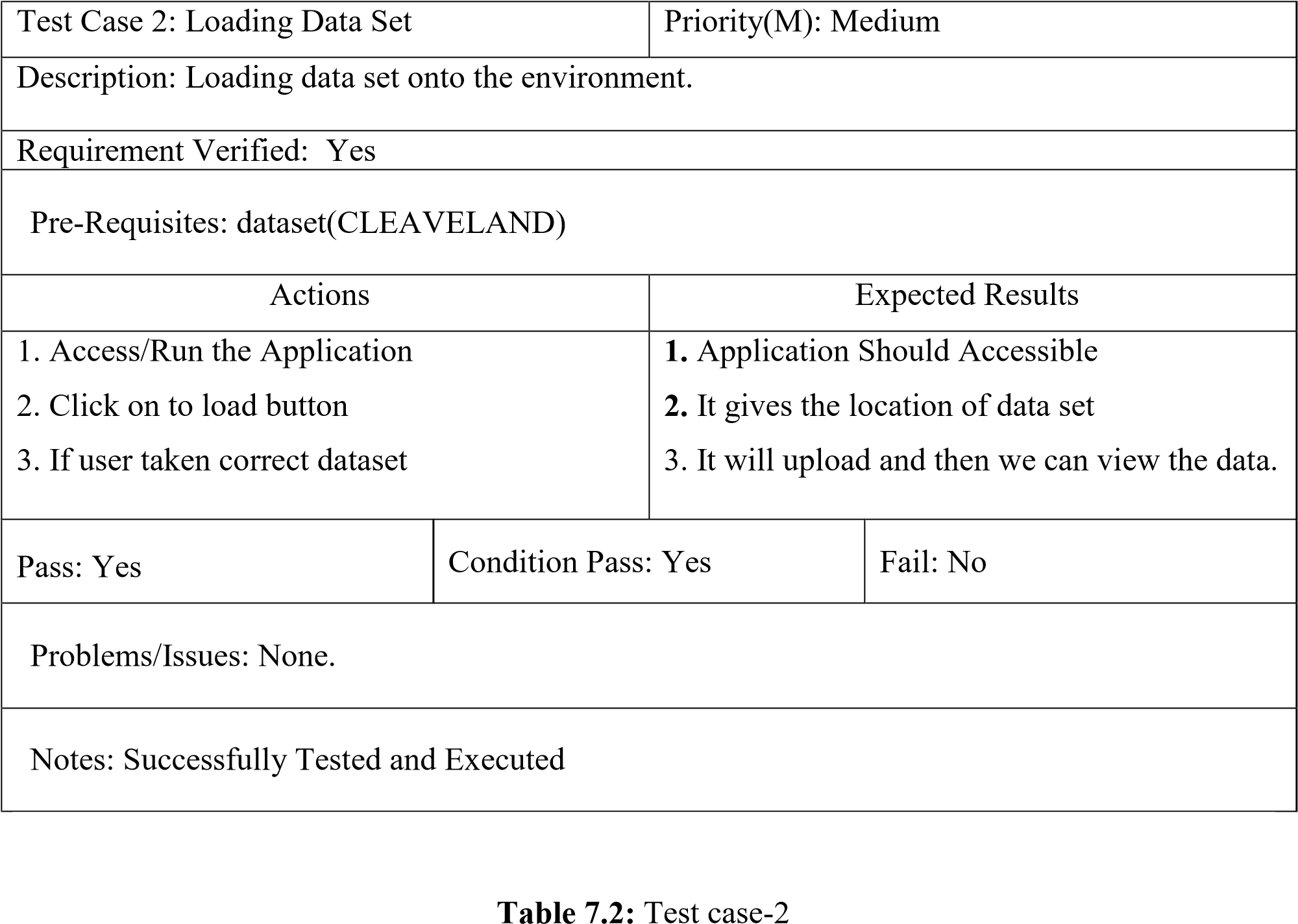
On the off chance that the client does not enter information in determined request, at that point the client will be incited with error messages. Error reduction is done to deal with the normal and sudden mistakes.

**7.4 Test Cases:**

It is a collection of circumstances or variables over which a tester will decide whether a scheme under examination fulfils necessities or works properly. The procedure of increasing test cases can as well aid to discover problems in the necessities or scheme of an appliance. A Test situation is any functionality that can be tested. It is also called Test Condition. As a tester, one may put themselves in the end user's shoes and find out the real-world scenarios of the Application under Test.

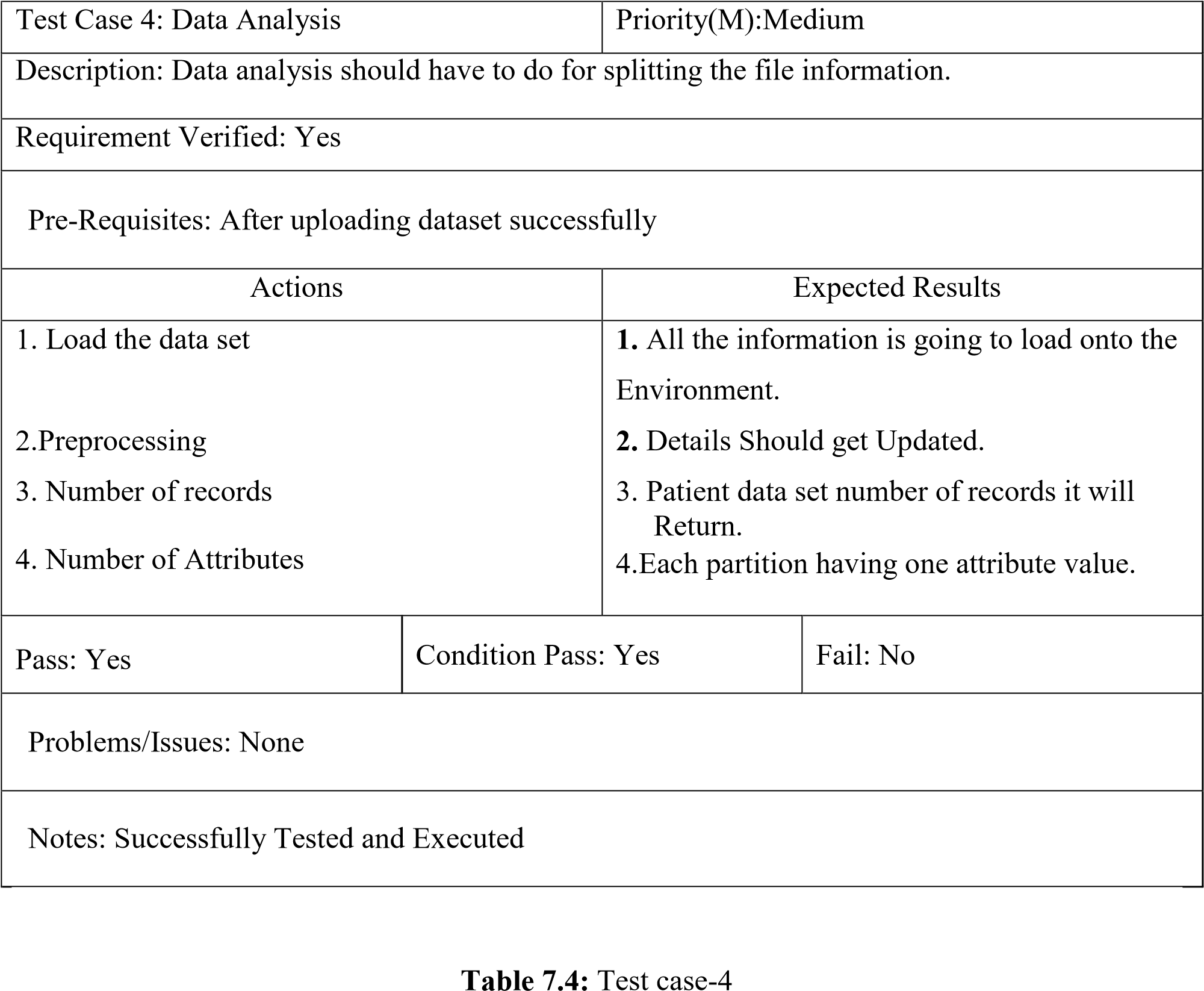
Below are the different test cases used to perform testing upon the functionality of the project?



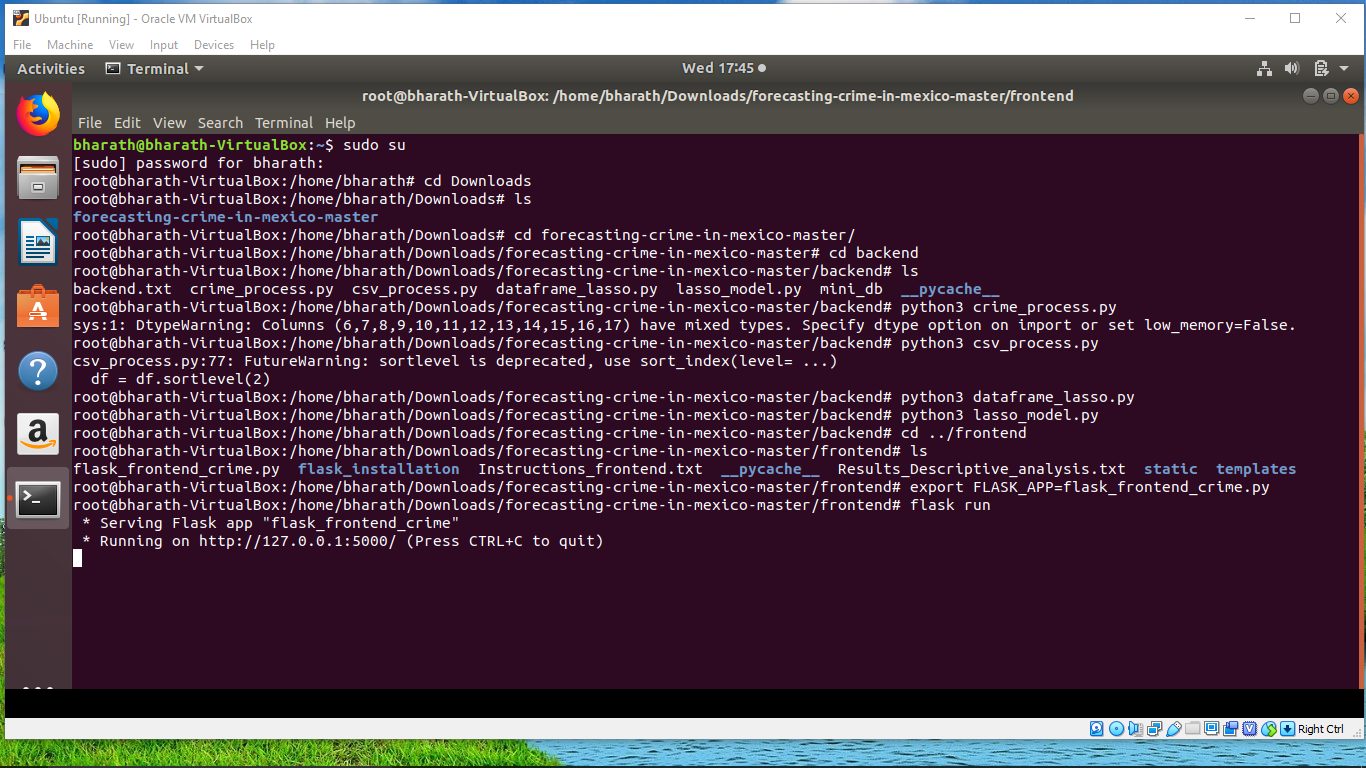


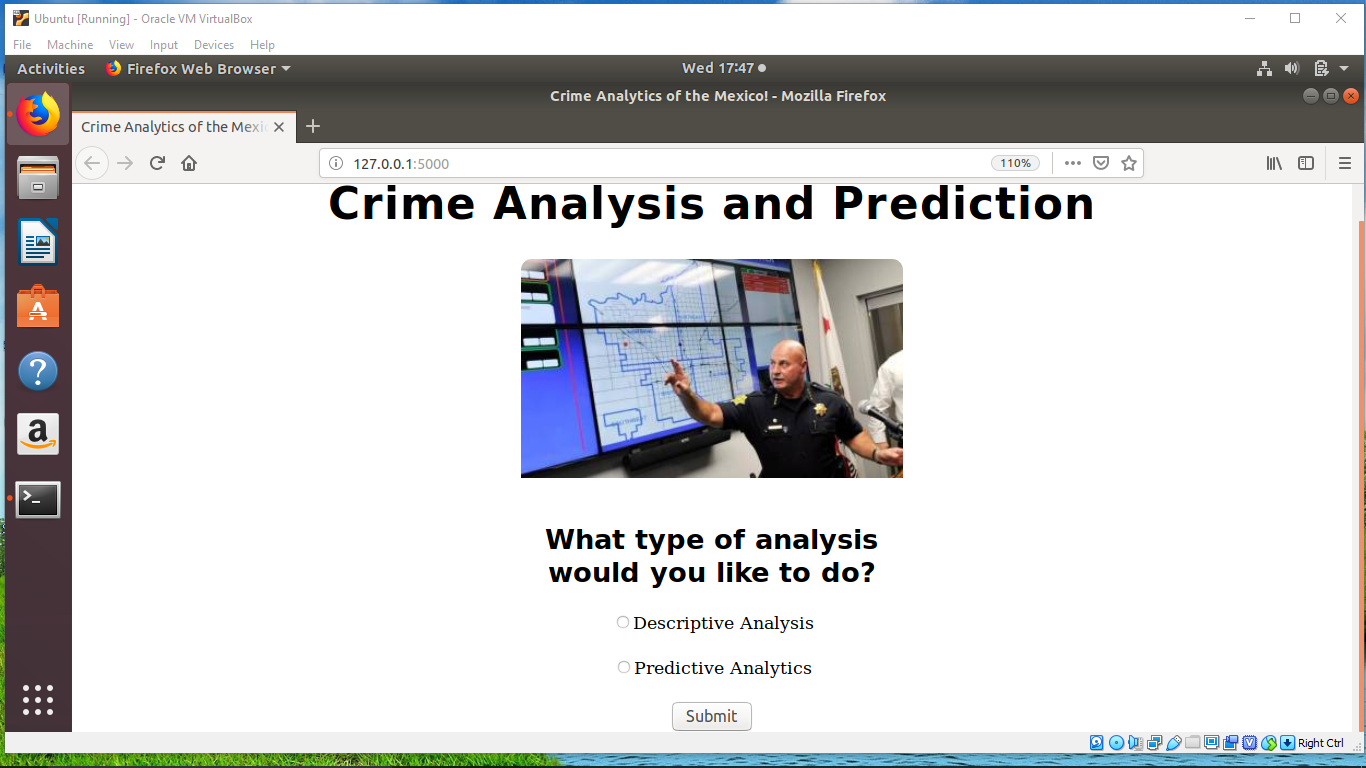
|  |  |  |  |
| --- | --- | --- | --- |
| Test Case 3: Data Analysis | | Priority(M): Medium | |
| Description: Data analysis should have to do for splitting the file information. | | | |
| Requirement Verified: Yes | | | |
| Pre-Requisites: After uploading dataset successfully | | | |
| Actions | | Expected Results | |
| 1. Load the data set    2.Preprocessing   1. Number of records      1. Number of Attributes | | 1. All the information is going to load onto the Environment. 2. Details Should get Updated. 3. Patient data set number of records it will Return.     4.Each partition having one attribute value. | |
| Pass: Yes | Condition Pass: Yes | | Fail: No |
| Problems/Issues: Wrong dataset, so analysis is not possible. | | | |
| Notes: Successfully Tested and Executed | | | |

**Table 7.3:** Test case-3

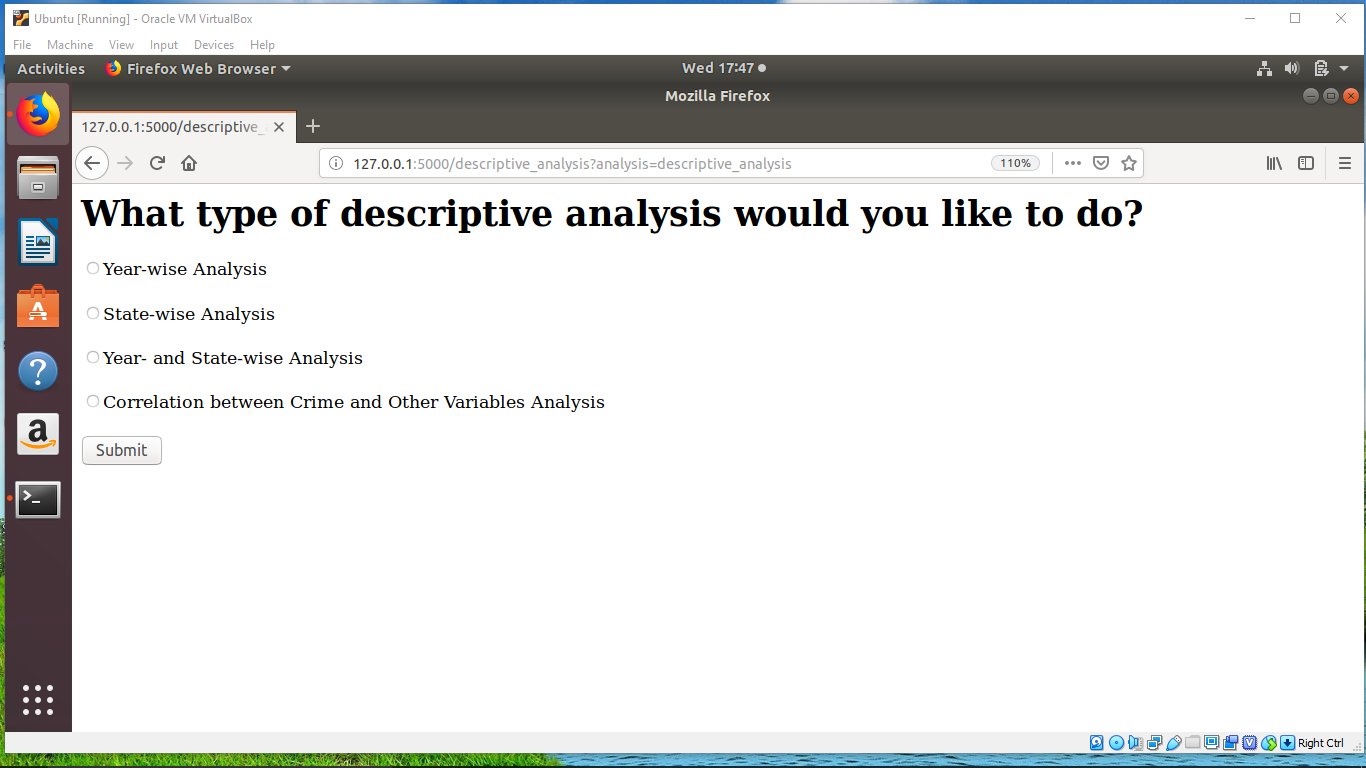


# 8. SCREENSHOTS

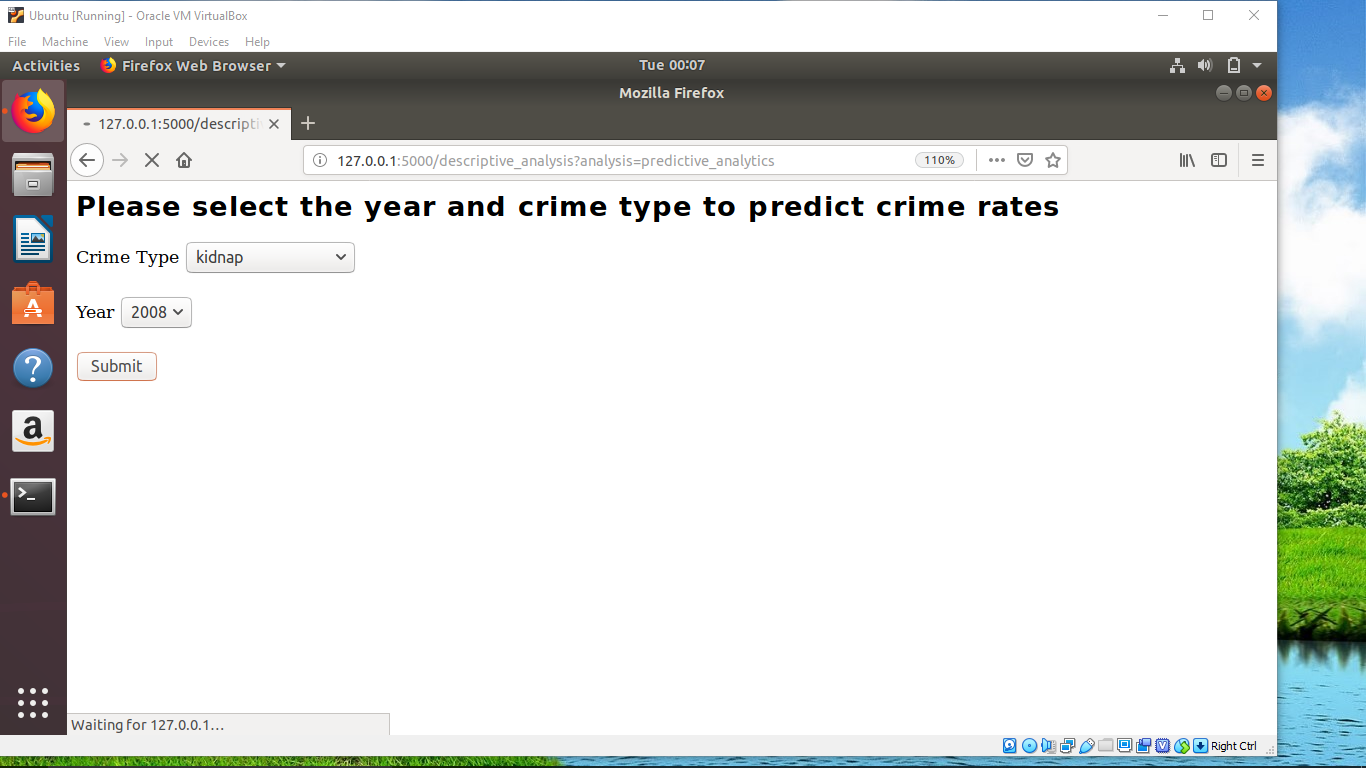


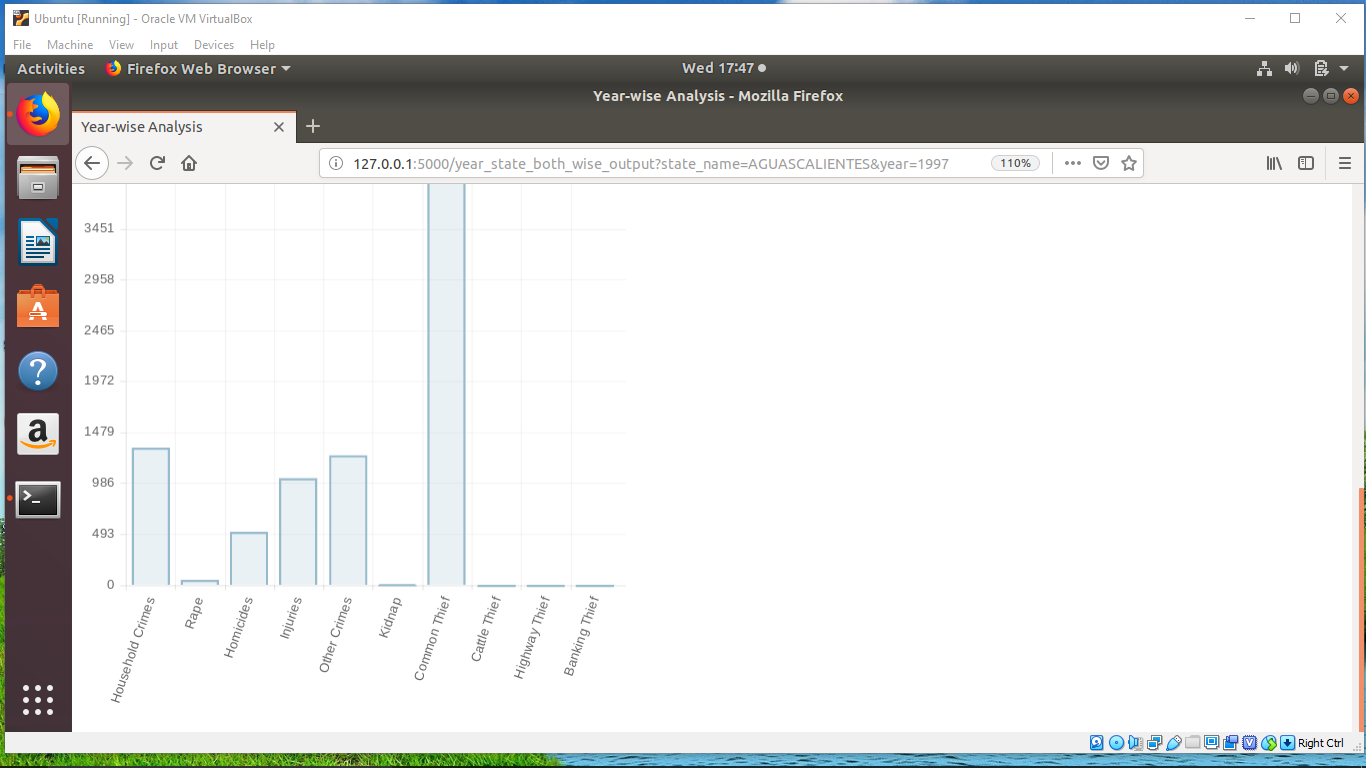


Home webpage



Analysis selection





Year wise selection

# 9. CONCLUSION

## 9.1 Conclusion:

Based on the results, it is evident that input data plays an important role in prediction along with prediction algorithms.

We have included data collection and cleaning, predictive analysis and descriptive analysis to display our results.

We found out the set of variables most closely correlated with crime, and predict with a certain level of accuracy where and when next crime will take place.

There are interesting insights that have come out from the Descriptive Analysis of the Crime in Mexico:

1. Baja California, Mexico and Ciuded de Mexico has a consistently highest rate of crime

1. The time period of 1999 to 2004 has a high rate of crime for majority of states in Mexico.

1. Most common crime types found:

—Common Thief

—Household crimes

—Injuries

—Other Crimes

1. Correlation Matrix:

--violent crimes are negatively correlated most with a high number of suspects of federal crimes

--violent crimes (robbery, homicides, kidnapping) are positively correlated most with a high number of cars and primary activities/employment/unemployment. This means (possibly) that most violent crimes in Mexico take place in urban/semi-urban areas. This also means that there is virtually little or less police control.

--non-violent crimes also have a similar pattern as of violent crimes

Most importantly, the above experiment not only helped us in predicting the outcome but also gave us valuable insights about the nature of data, which can be used in future to train our classifiers in a much better way.

## 9.2 Future Scope:

This system can be useful in combining Lasso with other techniques for a comparison matrix and same analysis can be used for smaller time and geographical units.

# 10. BIBLIOGRAPHY

## 10.1 References

1. https://www.w3schools.com/python/
2. https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/
3. https://catalog.data.gov/dataset?tags=crime
4. http://flask.pocoo.org/docs/0.12/installation/

[5] https://towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b

[6] https://www.quora.com/How-do-front-end-and-back-end-technologies-work-together

[7]https://www.predictiveanalyticsworld.com/book/pdf/PredictiveAnalyticsbyEric\_Siegel

1. <https://www.predictiveanalyticsworld.com/book/pdf/>
2. [Applied Predictive Modelling](http://www.amazon.com/gp/product/1461468485/ref=as_li_tl?ie=UTF8&camp=1789&creative=9325&creativeASIN=1461468485&linkCode=as2&tag=solutionsre04-20&linkId=DMEZNBIXOSQRTN6Q) by Max Kuhn and Kjell Johnson
3. <https://courses.cs.washington.edu/courses/cse599c1/13wi/slides/LARS-fusedlasso.pdf>
4. SQLite Forensics (2018) by Paul Sanderson.