

An Empirical Analysis of Machine Learning Algorithms for Crime Prediction Using Stacked Generalization

Submitted in partial fulfilment of the requirements for the award of
Master of Engineering degree in Computer Science and Engineering

by

SOLAPURI.SAI JAYANTH (40310004)



**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MAY - 2022

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of S. SAI JAYANTH
(Reg.no.40310004) who carried out the project entitle. '**An Empirical Analysis
of Machine Learning Algorithms for Crime Prediction Using
Stacked Generalization**' under my supervision. from NOV/2020 to MAY 2022.

Internal Guide

DR.N.SRINIVASAN,ME.,PhD

Head of the Department (Name in Capital letters with Seal & Signature)

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **S.SAI JAYANTH** hereby declare that the Project Report entitled
**“An Empirical Analysis of Machine Learning Algorithms for Crime
Prediction Using Stacked Generalization”** done by me under the guidance
of DR.N.SRINIVASAN,ME.,PhD.at Sathyabama Institute of Science and Technology,
is submitted in partial fulfilment of the requirements for the award of **Master of
Engineering in Computer Science and Engineering.**

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I thank the Board of Management of SATHYABAMA for their kind encouragement in doing this project and completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing , **Dr.S.Vigneshwari M.E., Ph.D. and Dr.L.Lakshmanan M.E., Ph.D. , Head** of the Department of Computer Science and Engineering for providing me the necessary support and guidelines at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide for his valuable guidance, **DR.N.SRINIVASAN,ME.,PhD** suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all teachers, technical staff and my fellow students of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The objective of this project is to tackle a vital issue in the society Crimes. Analyzing and examining of crimes happening in the world will give us a Broadview in understanding the crime regions and can be used to take necessary precautions to mitigate the crime rates. Identifying Crime patterns will allow us to tackle problems with unique approaches in specific crime category regions and improve more security measures in society. Current studies show the reason of increase in crime rates is more in areas that are economically backward. In few decades' property crime will be a target. The following approach involves predicting crimes classifying, pattern detection and visualization with effective tools and technologies. Use of past crime data trends helps us to correlate factors which might help understanding the future scope of crimes.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
I.	INTRODUCTION	1
1.1	Machine Learning	2
1.2	1.2 Machine Learning vs. Traditional Programming	3
1.3	How does Machine learning work?	3
1.4	Machine learning Algorithms and where they are used?	4
1.5	Application of Machine learning	7
1.6	Example of application of Machine Learning in Supply Chain	8
1.7	Example of Machine Learning Google Car	9
1.8	Deep Learning	9
1.9	Applications/ Examples of deep learning applications	10
II.	LITERATURE SURVEY	12
2.1	AIM OF THE PROJECT	15
III	SCOPE OF THE PRESENT INVESTIGATION	15
IV.	EXPERIMENTAL OR MATERIALS AND METHODS	16
4.1	BLOCK DIAGRAM	16
4.2	MODULES USED	16
4.3	ALGORITHMS USED	18
4.3.1.	SVM	18
4.3.1.1.	Works	18
4.3.2	KNN	20
4.3.3	LR	21
4.4	DIAGRAM USED IN OUR PROPOSED METHOD	21

4.5 SOFTWARE AND REQUIREMENTS	30
4.5.1 Python Overview	30
4.5.1.1 History of Python	30
4.5.1.2 Python Features	31
4.5.2 Tensor Flow Architecture	31
4.5.3 ANACONDA NAVIGATOR	33
4.5.4 TESTING	35
4.5.4.1 Testing Methods	34
4.5.5 System Requirements	36
V RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	37
VI SUMMARY AND CONCLUSIONS	70
S	

List of Abbreviations

ML-Machine Learning

AI-Artificial Intelligence

IP-Image Processing

CP-Crime Prediction

SVM-Support Vector Machine

RF-Random forest

LR-Linear Regression

GUI-Graphical User Interface

GPL-General Public License

SARSA-State-Action-Reward-State-Action

DDPG-Deep Deterministic Policy Gradient

UML-The Unified Modeling Language

LIST OF FIGURES

Figure 1: Machine Learning	3
Figure 2: Learning Phase diagram	4
Figure 3: Interference model of ML	5
Figure 4: Classification of ML	6
Figure 5: Block diagram of our proposed method	16
Figure 6: SVM diagram	23
Figure 7: Sequence Diagrams	24
Figure 8: Class Diagram	24
Figure 9: Activity Diagram	25
Figure 10: Colloboration Diagram	26
Figure 11: Use Case Diagram	27
Figure 12: Data Flow Diagram-Level0	28
Figure 13: Data Flow Diagram-Level 1	28
Figure 14: Data Flow Diagram-Level 2	29

Chapter I

Introduction

San Francisco was infamous for housing some of the world's most notorious criminals on the inescapable island of Alcatraz. Today, the city is known more for its tech scene than its criminal past. From Sunset to SOMA, and Marina to Excelsior, this project analysis 12 years of crime reports from across all of San Francisco's neighbourhoods to create a model that predicts the category of crime that occurred, given time and location. Recently, a lot of research and predictions have been attempted on how to curb crimes by various criminologists and researchers using different modeling and statistical tools. As the rate of crime is still on the hike, therefore, there is a potential need of some important research that can help the policy makers and the concerned department about challenges and issues in the area of crime prediction and control mechanisms. Skillset of human fails to keep track of criminal records, if handled manually. So, there is need for identifying in a novel way, which will help in analyzing crime related information. Analysis on crime prediction is currently based on two significant aspects, prediction of crime risk field and crime hotspot forecast. Data processing techniques are applied to facilitate this task. The expanded accessibility of computers and data innovations have empowered law authorization offices to incorporate broad databases with detailed information about major felonies, such as murder, rape, arson etc. In recent years, huge number of crimes is being reported in the world. Violence is a major crime in which a criminal threatens to use force on the victim. It refers to both crimes in which a violent act is the motive, such as murder or rape, robbery, as well as crimes in which violence is used as coercion. Crime may not necessarily, be initiated using weapons, depending on the jurisdiction, and violence crimes can range from murder to harassment. Typically, violence crimes include murders, robberies, rapes, attempt to murder, kidnapping, thefts, riots, dowry death, dowry atrocity etc. The rate of violent crimes is very high in 22 districts of Madhya Pradesh state and 23 districts of Rajasthan state in India; whereas the number of states is 34 and 25 respectively for the above-mentioned states

concerning the crimes against women. Commenting on the factual data in terms of the number of murders, it is reported that 17 districts listed among the poorest counties were in Maharashtra, with Andhra Pradesh and Rajasthan (having the next largest concentrations of violence crime districts). The crimes generally fall in the areas of violence criminal activities mostly against women. Similarly, defying public order as a crime was reported in 13, 12 and 11 districts of Rajasthan, Bihar and Tamil Nadu, respectively, out of total 100 districts.

The crime predictions are generally suggested by using machine learning techniques with respect to what percentage of future violence is possible in crimes. This research has been done for many years, but with some limited algorithms and small dataset. This research claims its novelty with the help of empirical analysis of machine learning and other contributions listed in this section. Though, machine learning models are widely used in crime prediction, but still despite of its expanding application and its gigantic potential, there are numerous regions, where the new procedures created in the zone of artificial intelligence have not been completely explored and has major drawbacks. The most common approaches which have reported achievable accuracy in machine learning classifiers are Random Tree Algorithm, K-Nearest Neighbor(KNN), Bayesian model, Support Vector Machine (SVM), Neural Network Among these algorithms, crime prediction technique is proposed by integrating a number of algorithms named as a crime prediction ensemble model using bagging and stacked ensemble techniques, reflecting the beauty of this research work. Ensemble model is a method for constructing a predictive model by combining multiple models to solve a single problem in order to improve predictive efficiency.

1.1 Machine Learning

Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results. Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights. Machine learning is closely related to data mining and Bayesian predictive modeling. The

machine receives data as input, use an algorithm to formulate answers. A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation. Machine learning is also used for a variety of task like fraud detection, predictive maintenance, portfolio optimization, automatize task and so on.

1.2 Machine Learning vs. Traditional Programming

Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.

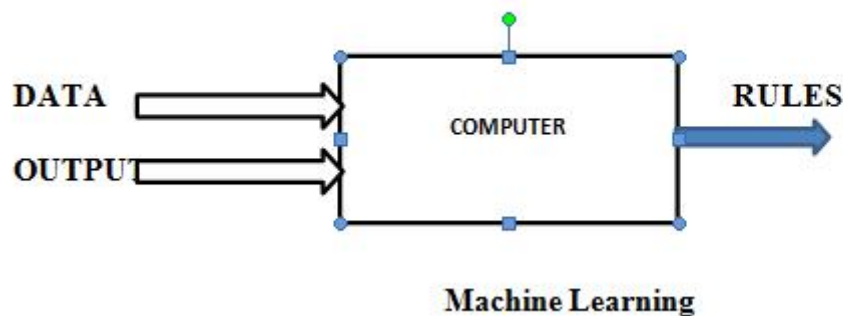


Figure 1 Machine Learning

1.3 How does Machine learning work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the

likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feed a previously unseen example, the machine has difficulties to predict. The core objective of machine learning is the learning and inference. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the data. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a feature vector. You can think of a feature vector as a subset of data that is used to tackle a problem. The machine uses some fancy algorithms to simplify the reality and transform this discovery into a model. Therefore, the learning stage is used to describe the data and summarize it into a model.

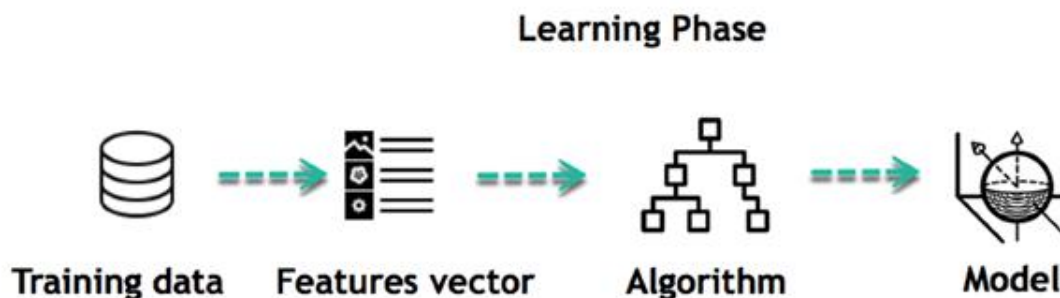


Figure 2 Learning Phase diagram

For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant.

Inferring

When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

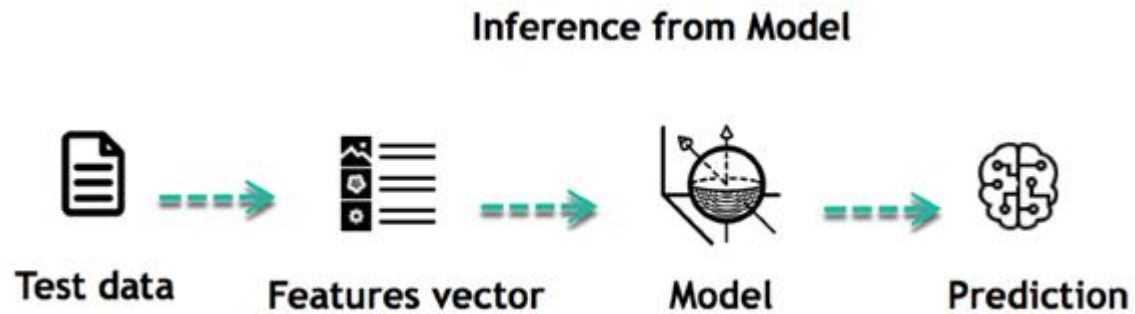


Figure 3 Inference model of ML

The life of Machine Learning programs is straightforward and can be summarized in the following points:

1. Define a question
2. Collect data
3. Visualize data
4. Train algorithm
5. Test the Algorithm
6. Collect feedback
7. Refine the algorithm
8. Loop 4-7 until the results are satisfying
9. Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data.

1.4 Machine learning Algorithms and where they are used?

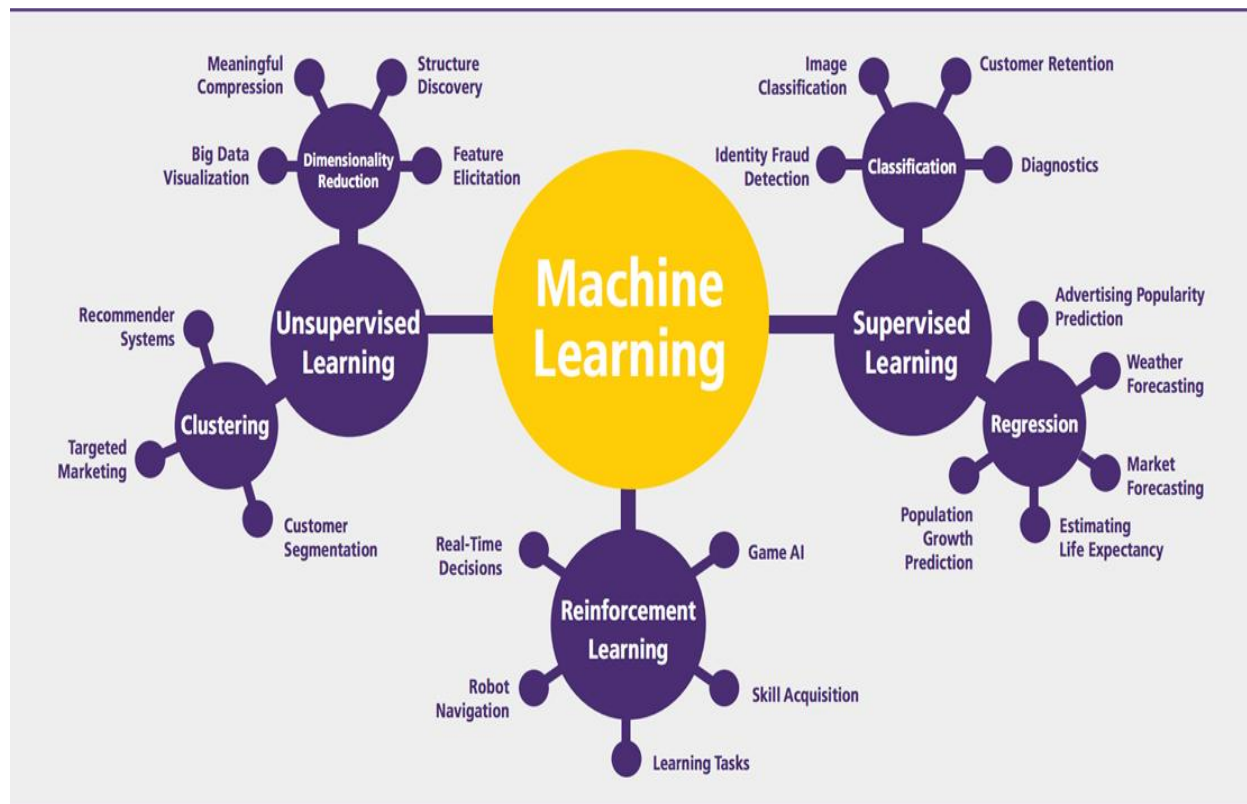


Figure 4 Classification of ML

Machine learning can be grouped into two broad learning tasks: Supervised and Unsupervised. There are many other algorithms

Supervised learning

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans.

Classification

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can only be male

or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female. The label can be of two or more classes. The above example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class)

Regression

When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of feature like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

Unsupervised learning

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns) You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you.

1.5 Application of Machine learning

Augmentation

- Machine learning, which assists humans with their day-to-day tasks, personally or commercially without having complete control of the output. Such machine learning is used in different ways such as Virtual Assistant, Data analysis, software solutions. The primary user is to reduce errors due to human bias.

Automation

- Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots performing the essential process steps in manufacturing plants.

Finance Industry

- Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.

Government organization

- The government makes use of ML to manage public safety and utilities. Take the example of China with the massive face recognition. The government uses Artificial intelligence to prevent jaywalker.

Healthcare industry

- Healthcare was one of the first industry to use machine learning with image detection.

Marketing

- Broad use of AI is done in marketing thanks to abundant access to data. Before the age of mass data, researchers develop advanced mathematical tools like Bayesian analysis to estimate the value of a customer. With the boom of data, marketing department relies on AI to optimize the customer relationship and marketing campaign.

1.6 Example of application of Machine Learning in Supply Chain

Machine learning gives terrific results for visual pattern recognition, opening up many potential applications in physical inspection and maintenance across the entire supply chain network. Unsupervised learning can quickly search for comparable patterns in the diverse dataset. In turn, the machine can perform quality inspection throughout the logistics hub, shipment with damage and wear. For instance, IBM's Watson platform can determine shipping container damage. Watson combines visual and systems-based data to track, report and make recommendations in real-time.

In past year stock manager relies extensively on the primary method to evaluate and forecast the inventory. When combining big data and machine learning, better forecasting techniques have been implemented (an improvement of 20 to 30 % over traditional forecasting tools). In term of sales, it means an increase of 2 to 3 % due to the potential reduction in inventory costs.

1.7 Example of Machine Learning Google Car

For example, everybody knows the Google car. The car is full of lasers on the roof which are telling it where it is regarding the surrounding area. It has radar in the front, which is informing the car of the speed and motion of all the cars around it. It uses all of that data to figure out not only how to drive the car but also to figure out and predict what potential drivers around the car are going to do. What's impressive is that the car is processing almost a gigabyte a second of data.

1.8 Deep Learning

Deep learning is a computer software that mimics the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural networks. The machine uses different layers to learn from the data. The depth of the model is represented by the number of layers in the model. Deep learning is the new state of the art in term of AI. In deep learning, the learning phase is done through a neural network.

Reinforcement Learning

Reinforcement learning is a subfield of machine learning in which systems are trained by receiving virtual "rewards" or "punishments," essentially learning by trial and error. Google's DeepMind has used reinforcement learning to beat a human champion in the Go games. Reinforcement learning is also used in video games to improve the gaming experience by providing smarter bot.

One of the most famous algorithms are

- Q-learning
- Deep Q network
- State-Action-Reward-State-Action (SARSA)
- Deep Deterministic Policy Gradient (DDPG)

1.9 Applications/ Examples of deep learning applications

AI in Finance: The financial technology sector has already started using AI to save time, reduce costs, and add value. Deep learning is changing the lending industry by using more robust credit scoring. Credit decision-makers can use AI for robust credit lending applications to achieve faster, more accurate risk assessment, using machine intelligence to factor in the character and capacity of applicants. Underwrite is a Fintech company providing an AI solution for credit makers company. underwrite.ai uses AI to detect which applicant is more likely to pay back a loan. Their approach radically outperforms traditional methods.

AI in HR: Under Armour, a sportswear company revolutionizes hiring and modernizes the candidate experience with the help of AI. In fact, Under Armour Reduces hiring time for its retail stores by 35%. Under Armour faced a growing popularity interest back in 2012. They had, on average, 30000 resumes a month. Reading all of those applications and begin to start the screening and interview process was taking too long. The lengthy process to get people hired and on-boarded impacted Under Armour's ability to have their retail stores fully staffed, ramped and ready to operate. At that time, Under Armour had all of the 'must have' HR technology in place such as transactional solutions for

sourcing, applying, tracking and onboarding but those tools weren't useful enough. Under armour choose HireVue, an AI provider for HR solution, for both on-demand and live interviews. The results were bluffing; they managed to decrease by 35% the time to fill. In return, the hired higher quality staffs.

AI in Marketing: AI is a valuable tool for customer service management and personalization challenges. Improved speech recognition in call-center management and call routing as a result of the application of AI techniques allows a more seamless experience for customers. For example, deep-learning analysis of audio allows systems to assess a customer's emotional tone. If the customer is responding poorly to the AI chatbot, the system can be rerouted the conversation to real, human operators that take over the issue. Apart from the three examples above, AI is widely used in other sectors/industries.

CHAPTER II

LITERATURE REVIEW

Almehmadi, Z. Joudaki, et al., "Language usage on Twitter predicts crime rates,". Social networks 1 produce enormous quantity of data. Twitter, a micro blogging network, consists of over 230 million active users posting over 500 million tweets every day. We propose to analyze public data from Twitter to predict crime rates. Crime rates have increased in the past recent years. Although crime stoppers are utilizing various techniques to reduce crime rates, none of the previous approaches targeted utilizing the language usage (offensive vs. non-offensive) in Tweets as a source of information to predict crime rates. In this paper, we hypothesize that analyzing the language usage in tweets is a valid measure to predict crime rates in cities. Tweets were collected for a period of 3 months in the Houston and New York City by locking the collection by geographic longitude and latitude. Further, tweets regarding crime events in the two cities were collected for verification of the validity of the prediction algorithm. We utilized Support Vector Machine (SVM) classifier to create a model of prediction of crime rates based on tweets. Finally, we report the validity of prediction algorithm in predicting crime rates in cities.

Z. Li, T. Zhang, et al, "Spatio-temporal pattern analysis and prediction for urban crime," This paper aims to analyze the intrinsic characteristics of urban crime in China by quantifying the crime data in the original case record. By comparing the predicted result of the crime situation with the observation, the intrinsic characteristic and its law are validated. Firstly, a quantitative method of case information based on Chinese description is developed, which can be used to transform the unstructured information in the case record to the security degree of the model. Secondly, analysis of the internal characteristics of the case, from the number of cases, the occurrence of time and location. Thirdly, an autoregressive integrated moving average model (ARIMA) based crime prediction model is presented for predict the situation of crime over a period of

time. The experimental results demonstrate that the predicted results are in line with the true values, and the predicted results show the same criminal characteristics.

S. Yadav, M. Timbadia, et al, "Crime pattern detection, analysis & prediction,". Crimes are a social irritation and cost our society deeply in several ways. Any research that can help in solving crimes quickly will pay for itself. About 10% of the criminals commit about 50% of the crimes. The system is trained by feeding previous years record of crimes taken from legitimate online portal of India listing various crimes such as murder, kidnapping and abduction, dacoits, robbery, burglary, rape and other such crimes. As per data of Indian statistics, which gives data of various crime of past 14 years (2001-2014) a regression model is created and the crime rate for the following years in various states can be predicted. We have used supervised, semi-supervised and unsupervised learning technique on the crime records for knowledge discovery and to help in increasing the predictive accuracy of the crime. This work will be helpful to the local police stations in crime suppression.

V. Ingilevich and S. Ivanov et al., "Crime rate prediction in the urban environment using social factors," The aim of this study is to compare different approaches to the problem of forecasting the number of crimes in different areas of the city. During this research we studied three types of predictive models: linear regression, logistic regression and gradient boosting. The predictive factors used in these models have been selected using the feature selection techniques. This approach allowed us to increase the accuracy of predictions and to avoid the model's overfitting. The obtained models were tested on criminal data of the city of Saint-Petersburg. We compared the results of model predictions and determined that gradient boosting is the most appropriate method for the problem of crime rate prediction in certain urban area.

L. G. A. Alves, H. V. Ribeiro et al., "Crime prediction through urban metrics and statistical learning," Understanding the causes of crime is a longstanding issue in researcher's agenda. While it is a hard task to extract causality from data, several linear models have been proposed to predict crime through the existing correlations between crime and urban metrics. However, because of non-Gaussian distributions and multicollinearity in urban indicators, it is common to find controversial conclusions

about the influence of some urban indicators on crime. Machine learning ensemble-based algorithms can handle well such problems. Here, we use a random forest regressor to predict crime and quantify the influence of urban indicators on homicides. Our approach can have up to 97% of accuracy on crime prediction, and the importance of urban indicators is ranked and clustered in groups of equal influence, which are robust under slightly changes in the data sample analyzed. Our results determine the rank of importance of urban indicators to predict crime, unveiling that unemployment and illiteracy are the most important variables for describing homicides in Brazilian cities. We further believe that our approach helps in producing more robust conclusions regarding the effects of urban indicators on crime, having potential applications for guiding public policies for crime control.

CHAPTER- III

AIM AND SCOPE OF OUR PROPOSED METHOD

3.1 AIM OF OUR PROJECT

The primary purpose of machine learning is to discover patterns in the user data and then make predictions based on these and intricate patterns for answering business questions and solving business problems.

3.2 SCOPE OF OUR PROJECT

Since machine learning needs you to know computer programming, statistics and data evaluation, the future scope of your machine learning career can also be in leadership roles in automation or analytics environments that use data science, big data analysis, AI integration of the System.

CHAPTER IV

EXPERIMENTAL ANALYSIS

In the Experimental results our project has shown the block diagram of our system. In that block diagram only what are steps can be used to show the results are evaluated.

4.1 BLOCK DIAGRAM

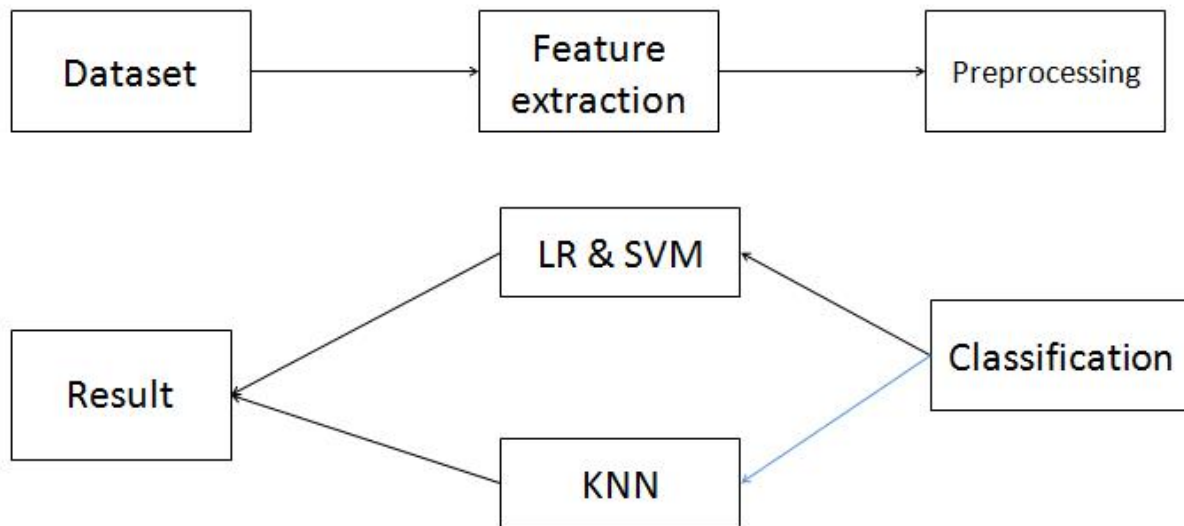


Figure 5 Block diagram of our proposed method

In our proposed method, the block shows some steps to show the results. Steps are Datasets collection, Pre Processing, Feature Extraction and Classification Part of the system.

4.2 MODULES USED

- Data Collection
- Data Pre-Processing
- Feature Extraction
- Evaluation Model

DATA COLLECTION

Data collection is the process of gathering and measuring information from countless different sources. In order to use the data we collect to develop practical artificial intelligence (AI) and machine learning solutions, it must be collected and stored in a way that makes sense for the business problem at hand. Data for which you already know the target answer is called labelled *data*.

DATA PRE-PROCESSING

Organize your selected data by formatting, cleaning and sampling from it.

Three common data pre-processing steps are:

- **Formatting:** The data you have selected may not be in a format that is suitable for you to work with. The data may be in a relational database and you would like it in a flat file, or the data may be in a proprietary file format and you would like it in a relational database or a text file.
- **Cleaning:** Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be anonymized or removed from the data entirely.
- **Sampling:** There may be far more selected data available than you need to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. You can take a smaller representative sample of the selected data that may be much faster for exploring and prototyping solutions before considering the whole dataset.

Feature Extraction

Features selection is done which can be used to build the model. The attributes used for feature selection are Block, Location, District, Community area, like dates, crime description, day of week, X, Y, Location.

Evaluation Model

In this module we implement the algorithm which one give best accuracy value. Based on that applying the algorithm it should calculate and give the accuracy. Here we are applying the algorithm and get the result on the form of graphical format.

4.3 ALGORITHM USED

Here, they are three types of algorithm are used. They are SVM, KNN and LR

4.3.1 SVM

Support Vector Machine (SVM) is a supervised machine learning algorithm capable of performing classification, regression and even outlier detection. The linear SVM classifier works by drawing a straight line between two classes. All the data points that fall on one side of the line will be labeled as one class and all the points that fall on the other side will be labeled as the second. Support Vector Machine or SVM is a supervised and linear Machine Learning algorithm most commonly used for solving classification problems and is also referred to as Support Vector Classification. There is also a subset of SVM called SVR which stands for Support Vector Regression which uses the same principles to solve regression problems. SVM also supports the kernel method also called the kernel SVM which allows us to tackle non-linearity.

4.3.1.1 WORKS

Just for the sake of understanding, we will leave the machines out of the picture for a minute. Now how would a human being like you and me classify a set of objects scattered on the surface of a table? Of course we will consider all their physical and visual characteristics and then identify based on our prior knowledge. We can easily identify and distinguish apples and oranges based on their colour, texture, shape etc. Now bringing back the machines, how would a machine identify an apple or an orange. Not surprisingly, it is based on the characteristics that we provide the machine with. It can be size, shape, weight etc. The more features we consider the easier it is to identify and distinguish both. For the time being, we will just focus on the weight and size(diameter) of apples and oranges. Now how would a machine using SVM, classify a

new fruit as either apple or orange just based on the data on the size and weights of some 20 apples and oranges that were observed and labelled? The below image depicts how.

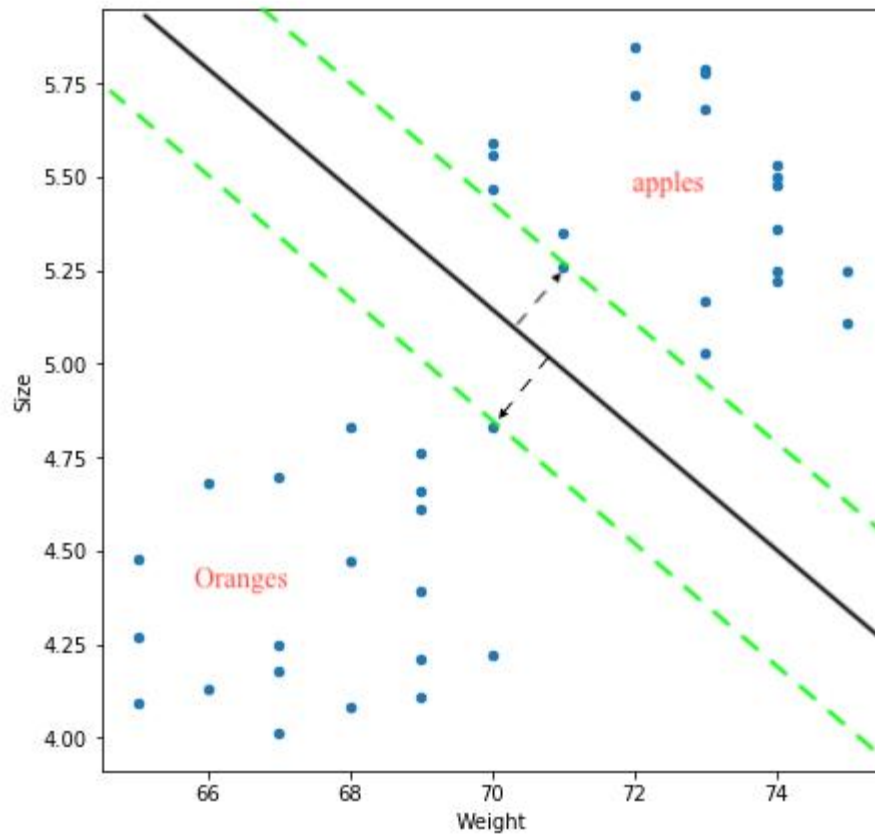


Figure 6 SVM diagram

The objective of SVM is to draw a line that best separates the two classes of data points. SVM generates a line that can cleanly separate the two classes. How clean, you may ask. There are many possible ways of drawing a line that separates the two classes, however, in SVM, it is determined by the **margins** and the **support vectors**. The margin is the area separating the two dotted green lines as shown in the image above. The more the margin the better the classes are separated. The support vectors are the data points through which each of the green lines passes through. These points are called support vectors as they contribute to the margins and hence the classifier itself.

These support vectors are simply the data points lying closest to the border of either of the classes which has a probability of being in either one. The SVM then generates a hyperplane which has the maximum margin, in this case the black bold line that separates the two classes which is at an optimum distance between both the classes. In case of more than 2 features and multiple dimensions, the line is replaced by a hyperplane that separates multidimensional spaces.

4.3.2 KNN

This algorithm is used to solve the classification model problems. K-nearest neighbor or K-NN algorithm basically creates an imaginary boundary to classify the data. When new data points come in, the algorithm will try to predict that to the nearest of the boundary line. Therefore, larger k value means smother curves of separation resulting in less complex models. Whereas, smaller k value tends to overfit the data and resulting in complex models. This algorithm is used to solve the classification model problems. K-nearest neighbor or K-NN algorithm basically creates an imaginary boundary to classify the data. When new data points come in, the algorithm will try to predict that to the nearest of the boundary line. Therefore, larger k value means smother curves of separation resulting in less complex models. Whereas, smaller k value tends to overfit the data and resulting in complex models.

4.3.3 LR

Regression analysis is one of the most important fields in statistics and machine learning. There are many regression methods available. Linear regression is one of them. Regression searches for relationships among variables. For example, you can observe several employees of some company and try to understand how their salaries depend on the features, such as experience, level of education, role, city they work in, and so on. This is a regression problem where data related to each employee represent one observation. The presumption is that the experience, education, role, and city are the independent features, while the salary depends on them.

Similarly, you can try to establish a mathematical dependence of the prices of houses on their areas, numbers of bedrooms, distances to the city center, and so on. Generally, in regression analysis, you usually consider some phenomenon of interest and have a number of observations. Each observation has two or more features. Following the assumption that (at least) one of the features depends on the others, you try to establish a relation among them. In other words, you need to find a function that maps some features or variables to others sufficiently well. The dependent features are called the dependent variables, outputs, **or** responses. The independent features are called the independent variables, inputs, **or** predictors. Regression problems usually have one continuous and unbounded dependent variable. The inputs, however, can be continuous, discrete, or even categorical data such as gender, nationality, brand, and so on.

4.4 DIAGRAM USED IN OUR PROPOSED METHOD

- UML Diagram
- Sequence Diagram
- Activity Diagrams
- Use-case diagram
- Class diagram
- Collaboration Diagram
- Dataflow Diagram

UML Diagram

The Unified Modeling Language (UML) is used to specify, visualize, modify, construct and document the artifacts of an object-oriented software intensive system under development. UML offers a standard way to visualize a system's architectural blueprints, including elements such as:

- actors
- business processes
- (logical) components

- activities
- programming language statements
- database schemas, and
- Reusable software components.

UML combines best techniques from data modeling (entity relationship diagrams), business modeling (work flows), object modeling, and component modeling. It can be used with all processes, throughout the software development life cycle, and across different implementation technologies. UML has synthesized the notations of the Booch method, the Object-modeling technique (OMT) and Object-oriented software engineering (OOSE) by fusing them into a single, common and widely usable modeling language. UML aims to be a standard modeling language which can model concurrent and distributed systems.

Sequence Diagrams:

Sequence Diagrams Represent the objects participating the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.

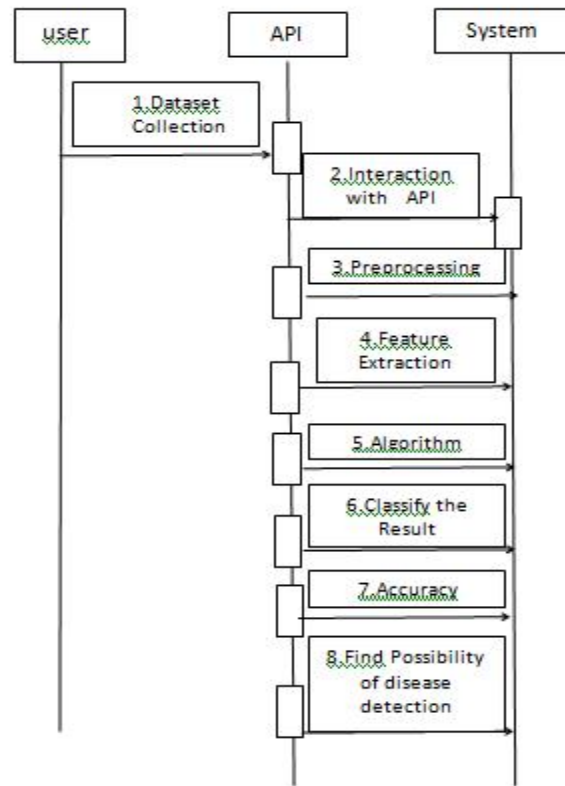


Figure 7 Sequence Diagrams

Class Diagram

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling.[1] The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

In the diagram, classes are represented with boxes that contain three compartments: The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized. The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase. The bottom compartment

contains the operations the class can execute. They are also left-aligned and the first letter is lowercase.

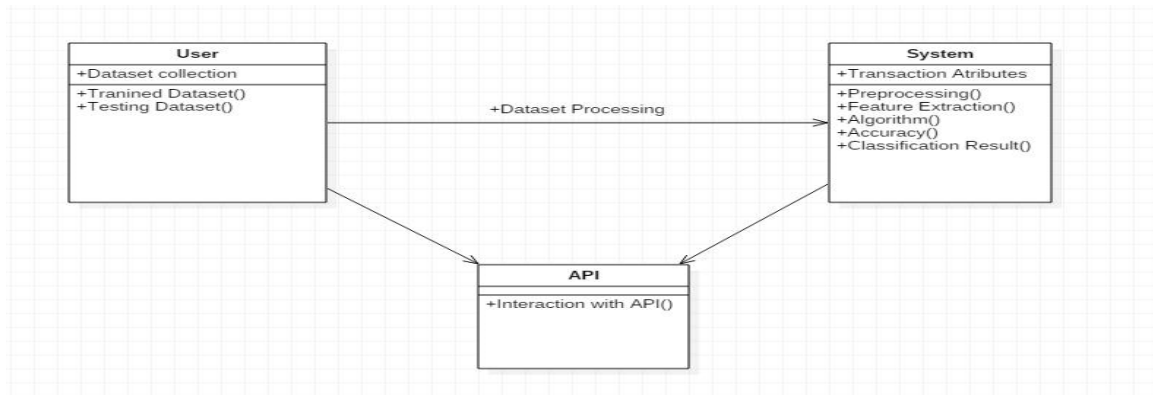


Figure 8 Class Diagram

Activity Diagram

Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

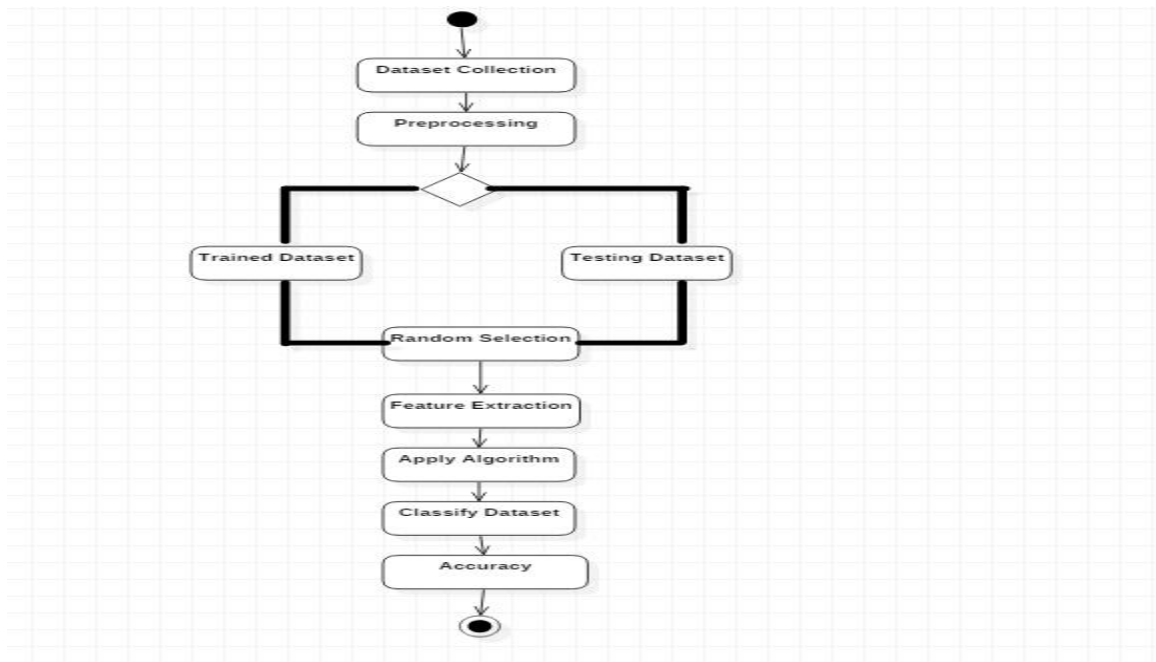


Figure 9 Activity Diagram

Colloboration Diagram

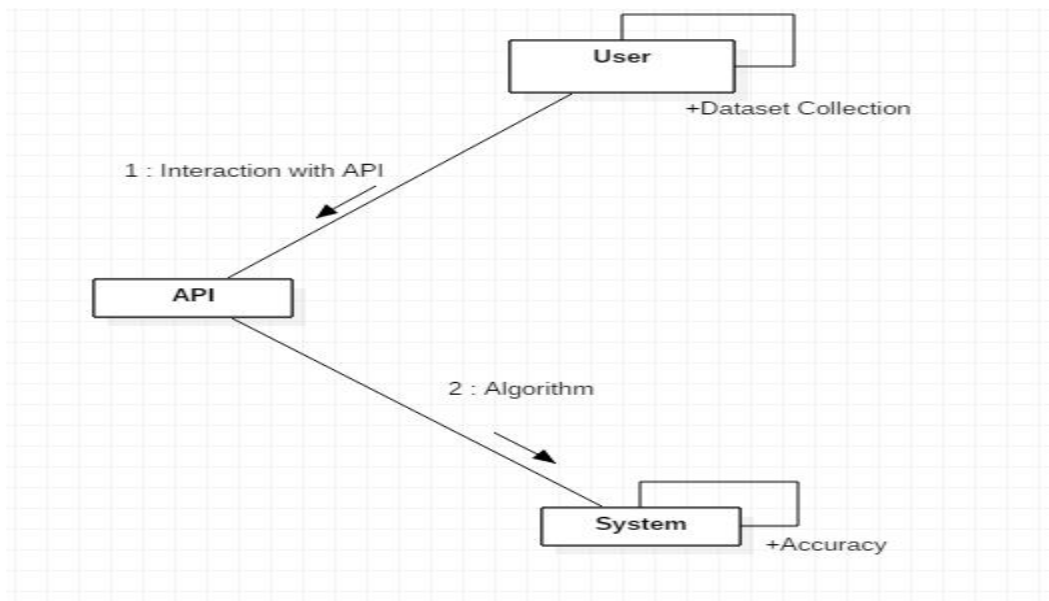


Figure 10 Colloboration Diagram

Use Case Diagram

- UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.
- UML was created by Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.
- OMG is continuously putting effort to make a truly industry standard.
- UML stands for Unified Modeling Language.
- UML is a pictorial language used to make software blue prints

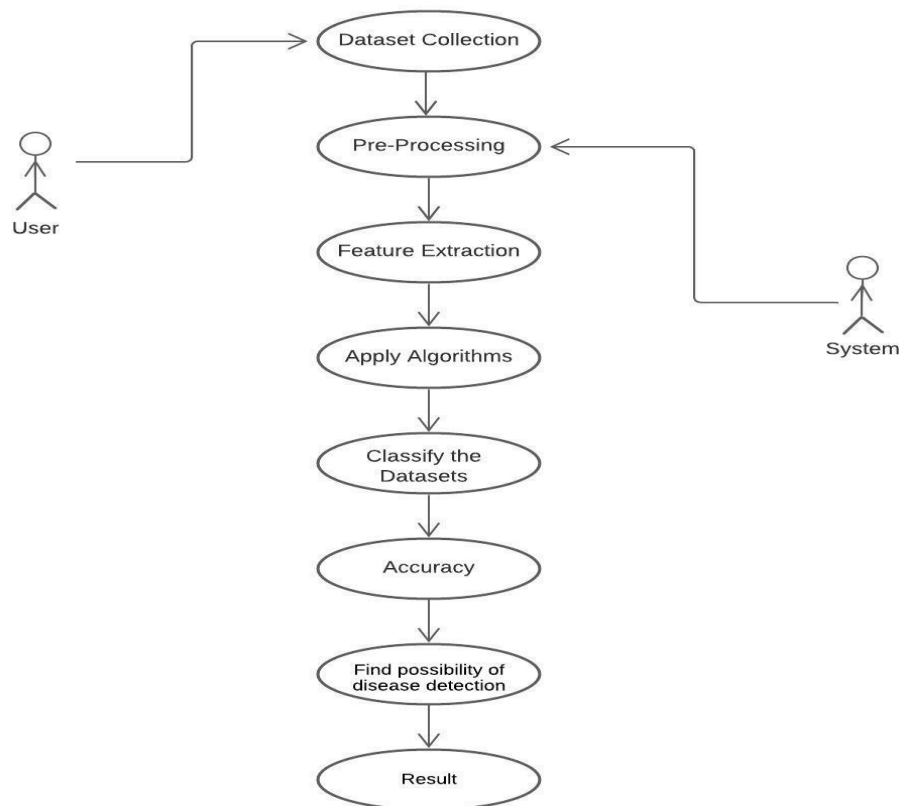


Figure 11 Use Case Diagram

Data Flow Diagram

In the data flow diagram, a diagram are categorized the level 0, level 1 and level 2.

Level 0

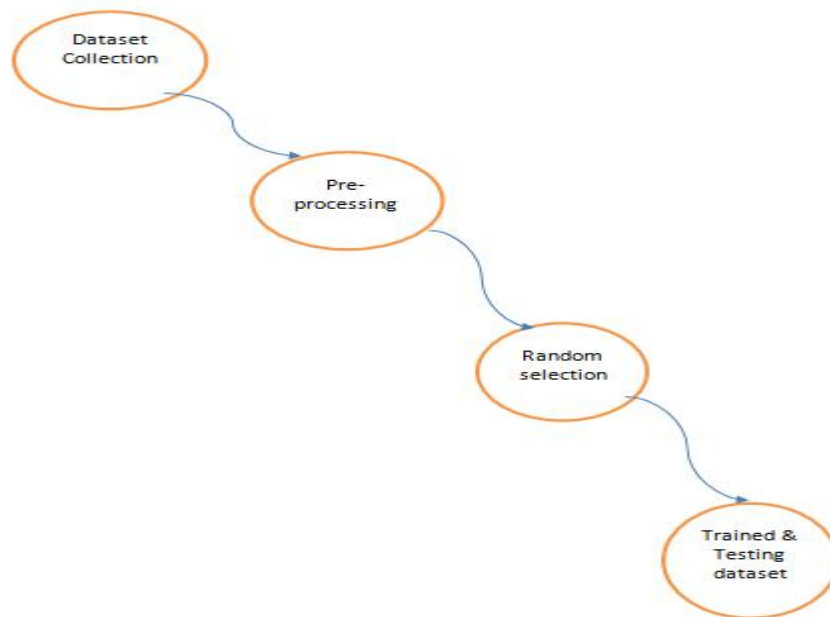


Figure 12 Data Flow Diagram-Level0

Level 1

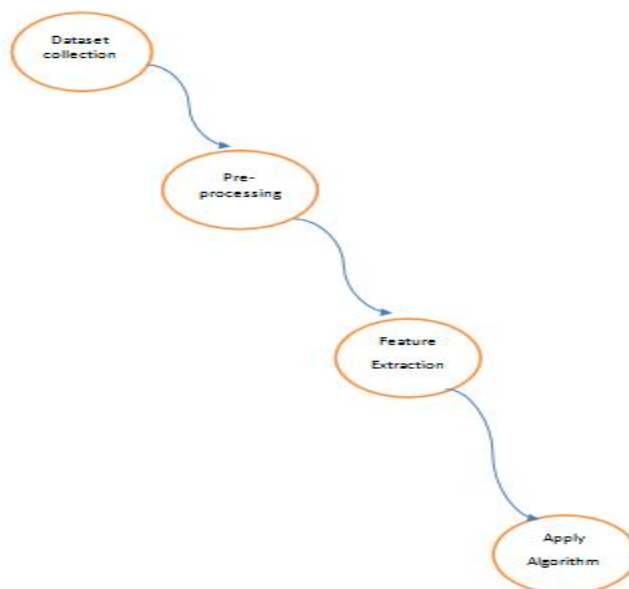


Figure 13 Data Flow Diagram-Level 1

Level 2

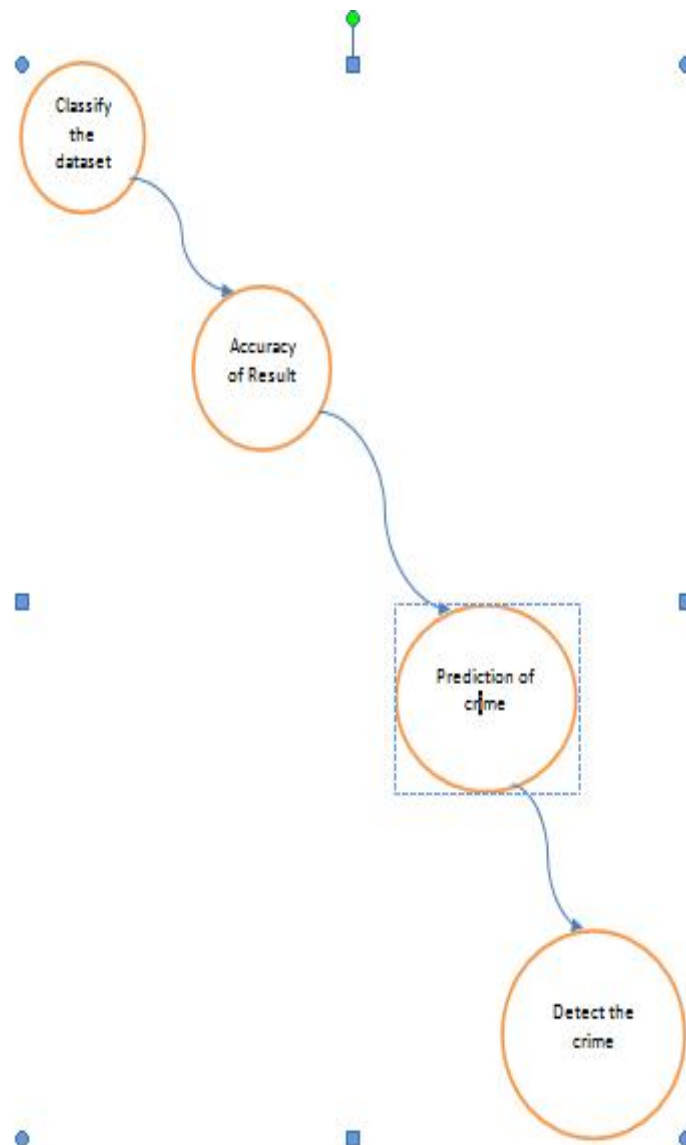


Figure 14 Data Flow Diagram-Level 2

4.5 SOFTWARE AND REQUIREMENTS

4.5.1 Python Overview

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted:** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- **Python is Interactive:** You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented:** Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language:** Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

4.5.1.1 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, Unix shell, and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

4.5.1.2 Python Features

Python's features include:

Easy-to-learn: Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.

Easy-to-read: Python code is more clearly defined and visible to the eyes.

Easy-to-maintain: Python's source code is fairly easy-to-maintain.

A broad standard library: Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.

Interactive Mode: Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.

Extendable: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.

Databases: Python provides interfaces to all major commercial databases.

GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries, and window systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

Scalable: Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below

- IT supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

- Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

4.5.2 TensorFlow Architecture

Tensor flow architecture works in three parts:

- Pre processing the data
- Build the model
- Train and estimate the model

It is called Tensor flow because it takes input as a multi-dimensional array, also known as **tensors**. You can construct a sort of **flowchart** of operations (called a Graph) that you want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the other end as output.

This is why it is called TensorFlow because the tensor goes in it flows through a list of operations, and then it comes out the other side.

Where can Tensor flow run?

TensorFlow can hardware, and software requirements can be classified into

Development Phase: This is when you train the mode. Training is usually done on your Desktop or laptop.

Run Phase or Inference Phase: Once training is done Tensorflow can be run on many different platforms. You can run it on

- Desktop running Windows, macOS or Linux
- Cloud as a web service
- Mobile devices like iOS and Android

You can train it on multiple machines then you can run it on a different machine, once you have the trained model. The model can be trained and used on GPUs as well as CPUs. GPUs were initially designed for video games. In late 2010, Stanford researchers found that GPU was also very good at matrix operations and algebra so that it makes them very fast for doing these kinds of calculations. Deep learning relies on a lot of matrix multiplication. TensorFlow is very fast at computing the matrix multiplication because it is written in C++. Although it is implemented in C++, TensorFlow can be accessed and controlled by other languages mainly, Python. Finally, a significant feature of Tensor Flow is the Tensor Board. The Tensor Board enables to monitor graphically and visually what TensorFlow is doing.

List of Prominent Algorithms supported by TensorFlow

- Linear regression: `tf.estimator.LinearRegressor`
- Classification :`tf.Estimator.Linear Classifier`
- Deep learning classification: `tf.estimator.DNN Classifier`
- Booster tree regression: `tf.estimator.BoostedTreesRegressor`
- Boosted tree classification: `tf.estimator.BoostedTreesClassifier`

4.5.3 ANACONDA NAVIGATOR

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, mac OS and Linux.

Why use Navigator?

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages, and use multiple environments to separate these different versions.

The command line program conda is both a package manager and an environment manager, to help data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages and update them, all inside Navigator.

WHAT APPLICATIONS CAN I ACCESS USING NAVIGATOR?

The following applications are available by default in Navigator:

- Jupyter Lab
- Jupyter Notebook
- QT Console
- Spyder
- VS Code
- Glue viz
- Orange 3 App
- Rodeo
- RStudio

Advanced conda users can also build your own Navigator applications”

How can I run code with Navigator? The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and execute your code. You can also use Jupyter Notebooks the same way. Jupyter Notebooks are an increasingly popular system that combine your code, descriptive text, output, images and interactive interfaces into a single notebook file that is edited, viewed and used in a web browser. What’s new in 1.9?

- Add support for **Offline Mode** for all environment related actions.
- Add support for custom configuration of main windows links.
- Numerous bug fixes and performance enhancements.

4.5.4 TESTING

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs.

Software Testing can also be stated as the process of validating and verifying that a software program/application/product:

- Meets the business and technical requirements that guided its design and Development.
- Works as expected and can be implemented with the same characteristics.

4.5.4.1 TESTING METHODS

Functional Testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- Functions: Identified functions must be exercised.
- Output: Identified classes of software outputs must be exercised.
- Systems/Procedures: system should work properly

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

4.5.5 System Requirements

Hardware:

- 1. Windows 7,8,10 64 bit
- 2. RAM 4GB

Software:

- 1. Python 2.7
- 2. Anaconda Navigator

CHAPTER V

RESULTS AND DISCUSSION

In the results and discussion part, we discussed the input program as well as the output screenshot of the system.

Program

```
{  
  
  "cells": [  
  
    {  
  
      "cell_type": "code",  
  
      "execution_count": 1,  
  
      "metadata": {  
  
        "_cell_guid": "  
  
      "outputs": [  
  
        {  
  
          "name": "stdout",  
  
          "output_type": "stream",  
  
          "text": [  
  
            "  
  
          "source": [  
  
            "import numpy as np \n",  
  
            "import pandas as pd \n",
```

```

"import warnings\n",

"warnings.filterwarnings('ignore')\n",

"\n",

"train_data = pd.read_csv('train.csv')\n",

"\n",

"print(train_data.head())\n",

"\n",

"print(train_data.shape)\n"

]

},

{

"cell_type": "markdown",

"metadata": {

"_cell_guid": "fb5a7f44-a20f-c23a-e71f-daa3bcc25a93"

},

"source": [

"# Understanding the data\n",

"\n",

"It's important to understand all the columns before we move further. Train data has the following columns:\n",

"\n",

"1. Dates - timestamp of the crime incident\n",

```

"2. **Category** - category of the crime incident (only in train.csv). This is the target variable you are going to predict.\n",

"3. **Descript** - detailed description of the crime incident (only in train.csv)\n",

"4. **DayOfWeek** - the day of the week\n",

"5. **PdDistrict** - name of the Police Department District\n",

"6. **Resolution** - how the crime incident was resolved (only in train.csv)\n",

"7. **Address** - the approximate street address of the crime incident \n",

"8. **X - Longitude**\n",

"9. **Y - Latitude**"

]

},

{

"cell_type": "code",

"execution_count": 2,

"metadata": {

"_cell_guid": "394231b7-4edc-f939-c889-2d887de413b8"

},

"outputs": [

{

"name": "stdout",

"output_type": "stream",

"text": [

```

["WARRANTS" 'OTHER OFFENSES' 'LARCENY/THEFT' 'VEHICLE THEFT' 'VANDALISM'\n",

" 'NON-CRIMINAL' 'ROBBERY' 'ASSAULT' 'WEAPON LAWS' 'BURGLARY'\n",

" 'SUSPICIOUS OCC' 'DRUNKENNESS' 'FORGERY/COUNTERFEITING' 'DRUG/NARCOTIC'\n",

" 'STOLEN PROPERTY' 'SECONDARY CODES' 'TRESPASS' 'MISSING PERSON' 'FRAUD'\n",

" 'KIDNAPPING' 'RUNAWAY' 'DRIVING UNDER THE INFLUENCE'\n",

" 'SEX OFFENSES FORCIBLE' 'PROSTITUTION' 'DISORDERLY CONDUCT' 'ARSON'\n",

" 'FAMILY OFFENSES' 'LIQUOR LAWS' 'BRIBERY' 'EMBEZZLEMENT' 'SUICIDE'\n",

" 'LOITERING' 'SEX OFFENSES NON FORCIBLE' 'EXTORTION' 'GAMBLING'\n",

" 'BAD CHECKS' 'TREA' 'RECOVERED VEHICLE' 'PORNOGRAPHY/OBSCENE MAT']\n"

]

}

],

"source": [

"target = train_data[\"Category\"].unique()\n",

"print(target)"

]

},

{

"cell_type": "code",

"execution_count": 3,

"metadata": {

```

"_cell_guid": "c6cfe480-4979-4458-c85b-1e6114d6b225"

},

"outputs": [

{

"name": "stdout",

"output_type": "stream",

"text": [

" Id Dates DayOfWeek PdDistrict Address \\n",

"0 0 2015-05-10 23:59:00 Sunday BAYVIEW 2000 Block of THOMAS AV \\n",

"1 1 2015-05-10 23:51:00 Sunday BAYVIEW 3RD ST / REVERE AV \\n",

"2 2 2015-05-10 23:50:00 Sunday NORTHERN 2000 Block of GOUGH ST \\n",

"3 3 2015-05-10 23:45:00 Sunday INGLESIDE 4700 Block of MISSION ST \\n",

"4 4 2015-05-10 23:45:00 Sunday INGLESIDE 4700 Block of MISSION ST \\n",

"\\n",

" X Y \\n",

"0 -122.399588 37.735051 \\n",

"1 -122.391523 37.732432 \\n",

"2 -122.426002 37.792212 \\n",

"3 -122.437394 37.721412 \\n",

"4 -122.437394 37.721412 \\n",

"(884262, 7)\\n"

```

    ]

    }

    ],

    "source": [

        "test_data = pd.read_csv('test.csv')\n",

        "print(test_data.head())\n",

        "print(test_data.shape)"

    ]

    },

    {

        "cell_type": "code",

        "execution_count": 4,

        "metadata": {

            "_cell_guid": "bdeb24b8-ccb2-200a-95fc-18cc0a03e7e3"

        },

        "outputs": [],

        "source": [

            "data_dict = {}\n",

            "count = 1\n",

            "for data in target:\n",

            "    data_dict[data] = count\n",

```

```

"    count+=1\n",

"train_data[\"Category\"] = train_data[\"Category\"].replace(data_dict)\n",

"\n",

"#Replacing the day of weeks\n",

"data_week_dict = {\n",

"    \"Monday\": 1,\n",

"    \"Tuesday\":2,\n",

"    \"Wednesday\":3,\n",

"    \"Thursday\":4,\n",

"    \"Friday\":5,\n",

"    \"Saturday\":6,\n",

"    \"Sunday\":7\n",

"}\n",

"train_data[\"DayOfWeek\"] = train_data[\"DayOfWeek\"].replace(data_week_dict)\n",

"test_data[\"DayOfWeek\"] = test_data[\"DayOfWeek\"].replace(data_week_dict)\n",

"#District\n",

"district = train_data[\"PdDistrict\"].unique()\n",

"data_dict_district = {}\n",

"count = 1\n",

"for data in district:\n",

"    data_dict_district[data] = count\n",

```

```

" count+=1 \n",

"train_data[\\"PdDistrict\\"] = train_data[\\"PdDistrict\\"].replace(data_dict_district)\n",

"test_data[\\"PdDistrict\\"] = test_data[\\"PdDistrict\\"].replace(data_dict_district)"

]

},

{

"cell_type": "code",

"execution_count": 5,

"metadata": {

"_cell_guid": "716ec2a1-9099-a56e-2353-72be757571eb"

},

"outputs": [

{

"name": "stdout",

"output_type": "stream",

"4 2015-05-13 23:30:00      3  GRAND THEFT FROM LOCKED AUTO      3  \n",

"\n",

" PdDistrict  Resolution      Address      X  \\\n",

"0      1  ARREST, BOOKED      OAK ST / LAGUNA ST -122.425892  \n",

"1      1  ARREST, BOOKED      OAK ST / LAGUNA ST -122.425892  \n",

"2      1  ARREST, BOOKED  VANNESS AV / GREENWICH ST -122.424363  \n",

```

```

"
    },
    "outputs": [
        {
            "name": "stdout",
            "output_type": "stream",
            "text": [
                "Index(['Dates', 'Category', 'Descript', 'DayOfWeek', 'PdDistrict',\n",
                "      'Resolution', 'Address', 'X', 'Y'],\n",
                "      dtype='object')\n",
                "Index(['Id', 'Dates', 'DayOfWeek', 'PdDistrict', 'Address', 'X', 'Y'], dtype='object')\n",
                ]
            }
        ],
        "source": [
            "columns_train = train_data.columns\n",
            "print(columns_train)\n",
            "columns_test = test_data.columns\n",
            "print(columns_test)"
        ]
    },

```

```

{

"cell_type": "code",

"execution_count": 7,

"metadata": {

"_cell_guid": "94b46c54-4c15-c779-7602-9d401b1c5904"

},

"outputs": [

{

"name": "stdout",

"output_type": "stream",

"text": [

"Index(['Dates', 'Category', 'Descript', 'DayOfWeek', 'PdDistrict', 'Address',\n",

"      'X', 'Y'],\n",

"      dtype='object')\n"

]

},

],

"source": [

"cols = columns_train.drop(\"Resolution\")\n",

"print(cols)"

]

```

```

},

{

"cell_type": "code",

"execution_count": 8,

"metadata": {

"_cell_guid": "216a89f5-ae4f-f687-bfea-61963419ee92"

},

"outputs": [

{

"name": "stdout",

"output_type": "stream",

"0      1      OAK ST / LAGUNA ST -122.425892  37.774599  \n",

"1      1      OAK ST / LAGUNA ST -122.425892  37.774599  \n",

"2      1  VANNES AV / GREENWICH ST -122.424363  37.800414  \n",

"3      1  1500 Block of LOMBARD ST -122.426995  37.800873  \n",

"4      2  100 Block of BRODERICK ST -122.438738  37.771541  \n"

]

},

],

"source": [

"train_data_new = train_data[cols]\n",

```

```

"print(train_data_new.head())"

]

},

{

"cell_type": "code",

"execution_count": 9,

"metadata": {

"_cell_guid":

"outputs": [

{

"name": "stdout",

"output_type": "stream",

"text": [

"max    39.000000    7.000000   10.000000  -120.500000  \n",

"count 878049.000000  \n",

"mean    37.771020  \n",

"std     0.456893  \n",

"min     37.707879  \n",

"25%     37.752427  \n",

"50%     37.775421  \n",

"75%     37.784369  \n",

```

```

"max      90.000000 \n"

]

}

],

"source": [

"print(train_data_new.describe())\n"

]

},

{

"cell_type": "code",

"execution_count": 10,

"metadata": {

"_cell_guid": "71924a13-e272-0915-0b4a-0fea471bab19"

},

"outputs": [

{

"name": "stdout",

"output_type": "stream",

"text": [

"Category      1.000000\n",

"DayOfWeek     -0.016263\n",

```

```

"PdDistrict  0.007643\n",

"X          0.000147\n",

"Y          -0.005303\n",

"Name: Category, dtype: float64\n"

]

}

],

"source": [

"corr = train_data_new.corr()\n",

"print(corr[\"Category\"])"

]

},

{

"cell_type": "code",

"execution_count": 11,

"metadata": {

"_cell_guid":

"outputs": [

{

"name": "stdout",

"output_type": "stream",

```

```

"

"# KNN Algorithm"

]

},

{

"cell_type": "code",

"execution_count": 12,

"metadata": {

"_cell_guid": "81141b9a-748e-d0cd-1572-0f0e001d50fa"

},

"outputs": [],

"source": [

"features = [\"DayOfWeek\", \"PdDistrict\", \"X\", \"Y\"]\n",

"X_train = train_data[features]\n",

"y_train = train_data[\"Category\"]\n",

"X_test = test_data[features]\n",

"\n",

"from sklearn.neighbors import KNeighborsClassifier\n",

"knn = KNeighborsClassifier(n_neighbors=5)\n",

"knn.fit(X_train, y_train)\n",

"predictions = knn.predict(X_test)"

```

```

]

},

{

"cell_type": "code",

"execution_count": 13,

"metadata": {

"_cell_guid": "35b26280-3045-401a-d11c-a90e381b247c"

},

"outputs": [

{

"name": "stdout",

"output_type": "stream",

},

"source": [

"from collections import OrderedDict\n",

"data_dict_new = OrderedDict(sorted(data_dict.items()))\n",

"print(data_dict_new)\n",

""

]

},

{

```

```

"cell_type": "code",

"execution_count": 14,

"metadata": {

  "_cell_guid"  },

"outputs": [

  {

    "ename": "MemoryError",

    "value": "",

    ]

  },

  ],

"source": [

"result_dataframe = pd.DataFrame({\n",

"  \"Id\": test_data[\"Id\"]\n",

"})\n",

"for key,value in data_dict_new.items():\n",

"  result_dataframe[key] = 0\n",

"count = 0\n",

"for item in predictions:\n",

"  for key,value in data_dict.items():\n",

"    if(value == item):\n",

```

```

"        result_dataframe[key][count] = 1\n",

"    count+=1\n",

"result_dataframe.to_csv(\"submission_knn.csv\", index=False) "

]

},

{

"cell_type": "markdown",

"metadata": {

"_cell_guid"

"source": [

"# Logistic Regression"

]

},

{

"cell_type": "code",

"execution_count": null,

"metadata": {

"_cell_guid"  },

"outputs": [],

"source": [

"from sklearn.linear_model import LogisticRegression\n",

```

```

"lgr = LogisticRegression()\n",

"lgr.fit(X_train, y_train)\n",

"predictions = knn.predict(X_test)\n",

"\n",

"#print(type(predictions))\n",

"result_dataframe = pd.DataFrame({\n",

"    \"Id\": test_data[\"Id\"]\n",

"})\n",

"for key,value in data_dict_new.items():\n",

"    result_dataframe[key] = 0\n",

"count = 0\n",

"for item in predictions:\n",

"    for key,value in data_dict.items():\n",

"        if(value == item):\n",

"            result_dataframe[key][count] = 1\n",

"        count+=1\n",

"result_dataframe.to_csv(\"submission_logistic.csv\", index=False) \n"

]

},

{

"cell_type": "markdown",

```

```

"metadata": {},

"source": [

"# LogisticRegression"

]

},

{

"cell_type": "code",

"execution_count": null,

"metadata": {

"_cell_guid": "00498f52-ad77-a919-7065-f71412f4f680"

},

"outputs": [],

"source": [

"from sklearn.linear_model import LogisticRegression\n",

"log = LogisticRegression()\n",

"log.fit(X_train, y_train)\n",

"predictions = log.predict(X_test)\n",

"\n",

"for key,value in data_dict_new.items():\n",

"    result_dataframe[key] = 0\n",

"count = 0\n",

```

```

"for item in predictions:\n",

"    for key,value in data_dict.items():\n",

"        if(value == item):\n",

"            result_dataframe[key][count] = 1\n",

"        count+=1\n",

"result_dataframe.to_csv(\"submission_logistic.csv\", index=False) "

]

},

{

"cell_type": "code",

"execution_count": null,

"metadata": {},

"outputs": [],

"source": []

}

],

"metadata": {

"_change_revision": 9,

"_is_fork": false,

"kernelspec": {

"display_name": "Python 3",

```

```
"language": "python",

"name": "python3"

},

"language_info": {

"codemirror_mode": {

"name": "ipython",

"version": 3

},

"file_extension": ".py",

"mimetype": "text/x-python",

"name": "python",

"nbconvert_exporter": "python",

"pygments_lexer": "ipython3",

"version": "3.7.0"

}

},

"nbformat": 4,

"nbformat_minor": 1

}
```

OUTPUT RESULTS

[illegible][illegible]

WPS Office | What's New | submit_n_knn | submit_logistic | test.csv | train.csv | An Emulation | modul_t put | + | Sign in | Go Premium

Menu | Paste | Copy | Cut | Format Painter | Calibri | 11 | Bold | Italic | Underline | Text Color | Fill Color | Background Color | Merge and Center | Wrap Text | General | Conditional Formatting | Cell Style | AutoSum | Autofilter | Sort | Fill | Format | Rows and Columns | Worksheet | Freeze Panes | Find and Replace

fx Id

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
	Id	ARSON	ASSAULT	BAD CHECK	BRIBERY	BURGLARY	DISORDERL	DRIVING U	DRUG/NAR	DRUNKENN	EMBEZZLEM	EXTORTION	FAMILY OFF	FORGERY/C	FRAUD	GAMBLING	KIDNAPPIN	LARCENY/T	LIQUOR LA	LOITERING	MISSING
2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
5	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	11	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
16	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
19	17	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	26	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

submission_logistic

32°C Light rain | 7:40 PM | 4/23/2022

CHAPTER VI

CONCLUSION

In this project study, ML models machine learning algorithms (ensemble and simile), SVM, SVM-Random forest, SVM-stacking and Naive Bias, were designed and were implemented. Each predetermined factor was feed into a violent crime training dataset (murder, rape, robbery, etc.). We discovered a major conclusion after successfully training and validating simodels.

CHAPTER VII

REFERENCES

- [1] M. Cahill and G. Mulligan, "Using geographically weighted regression to explore local crime patterns," *Social Sci. Comput. Rev.*, vol. 25, no. 2, pp. 174–193, May 2007, doi: 10.1177/0894439307298925.
- [2] J. M. Caplan, L. W. Kennedy, and J. Miller, "Risk terrain modeling: Brokering criminological theory and GIS methods for crime forecasting," *Justice Quart.*, vol. 28, no. 2, pp. 360–381, Apr. 2011, doi: 10.1080/07418825.2010.486037.
- [3] A. Almeahmadi, Z. Joudaki, and R. Jalali, "Language usage on Twitter predicts crime rates," in *Proc. 10th Int. Conf. Secur. Inf. Netw.*, Oct. 2017, pp. 307–310, doi: 10.1145/3136825.3136854.
- [4] V. K. Borooah and N. Ireland, "Deprivation, violence, and conflict: An analysis of Naxalite activity in the districts of India," *Int. J. Conf. Violence*, vol. 2, no. 2, pp. 317–333, 2008, doi: 10.4119/UNIBI/ijcv.42.
- [5] A. Babakura, M. N. Sulaiman, and M. A. Yusuf, "Improved method of classification algorithms for crime prediction," in *Proc. Int. Symp. Biometrics Secur. Technol. (ISBAST)*, Aug. 2014, pp. 250–255, doi: 10.1109/ISBAST.2014.7013130.
- [6] P. Pławiak, M. Abdar, and U. R. Acharya, "Application of new deep genetic cascade ensemble of SVM classifiers to predict the Australian credit scoring," *Appl. Soft Comput.*, vol. 84, Nov. 2019, Art. no. 105740, doi: 10.1016/j.asoc.2019.105740.
- [7] Z. Li, T. Zhang, Z. Yuan, Z. Wu, and Z. Du, "Spatio-temporal pattern analysis and prediction for urban crime," in *Proc. 6th Int. Conf. Adv. Cloud Big Data (CBD)*, Aug. 2018, pp. 177–182, doi: 10.1109/CBD.2018.00040.
- [8] A. Almaw and K. Kadam, "Survey paper on crime prediction using ensemble approach," *Int. J. Pure Appl. Math.*, vol. 118, no. 8, pp. 133–139, 2018. [Online]. Available: <https://internal-pdf://107.93.182.66/18.pdf%0Ahttp://www.ijpam.eu>

- [9] T. B. Hyde, H. Dentz, S. A. Wang, H. E. Burchett, S. Mounier-Jack, and C. F. Mantel, "The impact of new vaccine introduction on immunization and health systems: A review of the published literature," *Vaccine*, vol. 30, no. 45, pp. 6347–6358, 2015, doi: 10.1016/j.vaccine.2012.08.029
- [10] S. Yadav, M. Timbadia, A. Yadav, R. Vishwakarma, and N. Yadav, "Crime pattern detection, analysis & prediction," in *Proc. Int. Conf. Electron., Commun. Aerospace. Technology. (ICECA)*, Apr. 2017, pp. 225–230, doi: 10.1109/ICECA.2017.8203676.
- [11] C. M. Bishop.: *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [12] W. Hart.: *Non-intrusive appliance load monitoring*. *Proceedings of the IEEE*, vol. 80, no. 12, 1992.
- [13] J. Moody et al.: *Fast learning in networks of locally tuned processing units*. *Neural Computation* 1 (2), , 692498, 2001. 281-294, 1989.
- [14] H. Murata et al.: *Applying Kernel Based Subspace classification to a Non-Intrusive Monitoring System for Household Electric Appliances*. *ICANN2001*
- [15] T. Onoda et al.: *Applying Support Vector Machines and Boosting to a Non-Intrusive Monitoring System for Household Electric Appliances with Inverters*. *NC'2000*, 2000.
- [16] D. E. Rumelhart et al.: *Learning Internal Representations by Error Propagation*. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. I: Foundations, 318-362, Bradford Books/MIT Press, Cambridge, MA., 1986.
- [17] V. N. Vapnik.: *The Nature of Statistical Learning Theory*. J. Springer, 1995.