ABSTRACT

Crime is an alarming aspect of our society, and its prevention is a vital task. Crime analysis is a well-organized way of detecting and examining patterns and trends in crime. It is of utmost importance to study reasons, consider different factors and determine the relationship among various crimes occurring and discover the best suitable methods to control crime. The primary objective of this project is to distinguish various crimes using clustering techniques based on the occurrences and regularity.

Data mining is used for analysis, investigation and check patterns in crimes. In this project, a clustering approach is used to analyze the crime data; the stored data is clustered using the K-Means algorithm. After the clustering, we can predict a crime based on its historical information using classification. This proposed system can indicate crime head which have a high probability of crime rate. Analyze crime to meet the law enforcement needs of a changing society analyze crime to understand the criminal behaviors.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ML	- -	Machine Learning
SVM	- -	Support Vector Machine
ANN	- -	Artificial Neural networks
KNN	- -	K-Nearest Neighbor
UML	- -	Unified Modeling Language
OMT	- -	Object Modeling Technique
OOSE	- -	Object Oriented Software Engineering
OMG	- -	Object Management Group
AI	- -	Artificial Intelligence
DDPG	- -	Deep Deterministic Policy Gradient
GPU	- -	Graphics Processing Unit
GPL	- -	General Public License
GUI	- -	Graphical User Interface
FIR	- -	First Information Report

TABLE OF CONTENTS

Contents **Page No.**

CHAPTER-1

INTRODUCTION

1.1 Introduction

Crimes are the significant threat to the humankind. There are many crimes that happens regular interval of time. Perhaps it is increasing and spreading at a fast and vast rate. Crimes happen from small village, town to big cities. Crimes are of different type – robbery, murder, rape, assault, battery, false imprisonment, kidnapping, homicide. Since crimes are increasing there is a need to solve the cases in a much faster way.

The crime activities have been increased at a faster rate and it is the responsibility of police department to control and reduce the crime activities. Crime prediction and criminal identification are the major problems to the police department as there are tremendous amount of crime data that exist. There is a need of technology through which the case solving could be faster.

Crime is increasing considerably day by day. Crime is among the main issues which is growing continuously in intensity and complexity. Crime patterns are changing constantly because of which it is difficult to explain behaviors in crime patterns. Crime is classified into various types like kidnapping, theft murder, rape etc. The law enforcement agencies collect the crime data information with the help of information technologies. But occurrence of any crime is naturally unpredictable and from previous searches it was found that various factors like poverty, employment affects the crime rate.

It is neither uniform nor random. With rapid increase in crime number, analysis of crime is also required. Crime analysis basically consists of procedures and methods that aims at reducing crime risk. It is a practical approach to identify and analyze crime patterns. But, major challenge for law enforcement agencies is to analyze escalating number of crime data efficiently and accurately. So, it becomes a difficult challenge for crime analysts to analyze such voluminous crime data without any computational support.

1.2 Motivation

To make crime prediction using the features present in the dataset. The dataset is extracted from the official sites. With the help of machine learning algorithm, using python as core we can predict the type of crime which will occur in a particular area.

1.3 Objective

The objective would be to train a model for prediction. The training would be done using the training data set which will be validated using the test dataset. Building the model will be done using better algorithm depending upon the accuracy. The K-means clustering and few classifications algorithm will be used for crime prediction. Visualization of dataset is done to analyze the crimes which may have occurred in the particular state. This work helps the law enforcement agencies to predict and detect crimes in India with improved accuracy and thus reduces the crime rate.

1.4 Organization of Thesis

The rest of the thesis is organized in the following manner:

Chapter 2 - Deals with Literature Survey. Here some basic concepts are explained.

Chapter 3 - Contains the proposed work and analysis.

Chapter 4 - Gives the detailed description of Modules using UML Diagrams.

Chapter 5 - Gives the domain specifications and applications.

Chapter 6 - Deals with the tensor flow and the technology used.

Chapter 7 - Gives the detailed description of tool used and implementation.

Chapter 8 - Deals with the Debugging process of the proposed system.

Chapter 9 - Here the results of the implemented modules along with the respective Screenshots are provided.

Chapter 10 - Here the Conclusions of the current application are provided.

Chapter 11 - Here the enhancement work for the future research are given.

Chapter 12 - Here the References, Text Books, Web Sites for this work are given.

CHAPTER-2

LITERATURE SURVEY

2.1 Literature Survey

Many researches have been done which address this problem of reducing crime and many crime-predictions algorithms has been proposed. The prediction accuracy depends upon on type of data used, type of attributes selected for prediction. In mobile network activity was used to obtain human behavioral data which was used to predict the crime hotspot in London with an accuracy of about 70% when predicting that whether a specific area in London city will be a hotspot for crime or not.

In data collected from various websites, newsletter was used for prediction and classification of crime using Naive Bayes algorithm and decision trees and found that former performed better.

In a thorough study of various crime prediction method like Support Vector Machine (SVM), Artificial neural networks (ANN) were done and concluded that there does not exist particular method which can solve different crime datasets problems.

In various supervised learning techniques, unsupervised learning technique on the crime records were done which address the connections between crime and crime pattern for the purpose of knowledge discovery which will help in increasing predictive accuracy of crime. In different approach for predicting like Data mining technique, Deep learning technique, Crime cast technique, Sentimental analysis technique were discussed and it was

found that every method has some cons and pros. Every method gives better result for a particular instance.

Clustering approaches were used for detection of crime and classification method were used for the prediction of crime. The K-Means clustering was implemented and their performance is evaluated on the basis of accuracy. On comparing the performance of different clustering algorithm DBSCAN gave result with highest accuracy and KNN classification algorithm is used for crime prediction. Hence, this system helps law enforcement agencies for accurate and improved crime analysis.

In a comparison of classification algorithms, Naïve Bayes and decision tree was performed with a data mining software, WEKA. The datasets for this study were obtained from US Census 1990. In the pattern of road accidents in Ethiopia were studied after taking into consideration various factors like the driver, car, road conditions etc. Different classification algorithms used were K-Nearest Neighbor, Decision tree and Naive Bayes on a dataset containing around 18000 datapoints. The prediction accuracy for all three methods was between 79% to 81%.

2.2 Existing system

Based on the previous year crime details in Indian states, its present statistical models through Weighted Moving Average, Functional Coefficient Regression and Arithmetic-Geometric Progression based prediction of the crime in coming years. Difference between actual records and our predicted values for both years gives the accuracy of the proposed approaches between the range 85% and 90%. In future, this work can be modified by using Machine Learning (ML) models for forecasting crime, as the data points will sufficiently

increase to apply ML models. This can also increase the accuracy of the predictions. Further, statistical modeling's methods can also be clubbed with ML models and then calculate weighted accuracy for a district, this can make the solution more robust.

2.3 Disadvantages

Although some approaches and some detection techniques are present like women safety security system in IoT and embedded, there was some accuracy problems.

2.3.1 DESIGN METHODOLOGY

An unsupervised machine learning model used for making clusters of crime type as crime head labels further a supervised machine learning model is trained by using crime head data. The model is trained to predict the probability that a new crime head that should be reported and we can avoid crimes to be occurred.

2.3.2 PROPOSED SYSTEM

There are many machine learning algorithms available to users that can be implemented on datasets. However, there are two major types of learning algorithms: supervised learning and unsupervised learning algorithms. Supervised learning algorithms work by inferring information or "the right answer" from labeled training data. The algorithms are given a particular attribute or set of attributes to predict. Data preprocessing process includes methods to remove any null values or infinite values which may affect the accuracy of the system. The main steps include Formatting, cleaning and sampling. Cleaning process is used for removal or fixing of some missing data there may be data that are incomplete.

2.3.3 Crimes Prediction ways

- To utilize the resources, identify the hotspots of crimes and allocate vigilante resources such as policeman, police cars, weapons etc. reschedule patrols according to the vulnerability of a place.
- Through that avoid crimes Ensure better civilization through avoiding happening crimes such as murder, rapes, thefts, drug, smugglings etc.

2.3.4 System Architecture

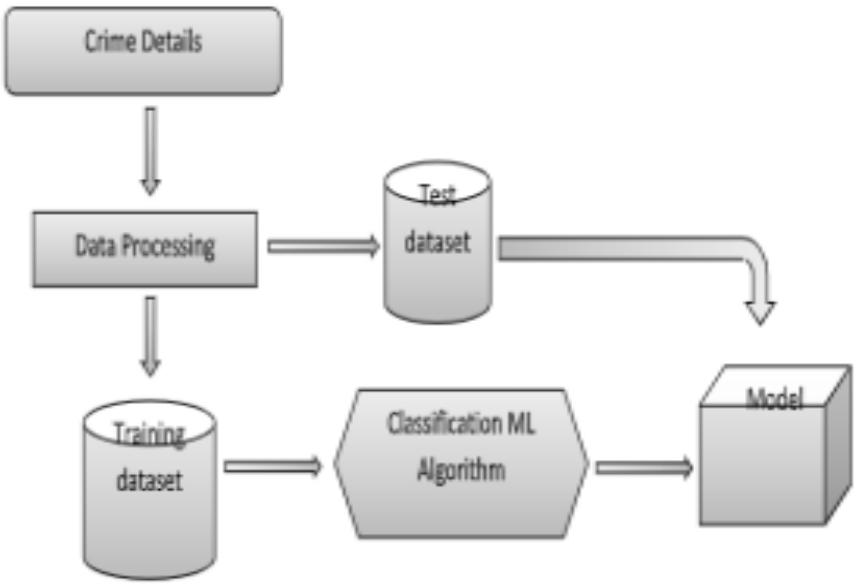


Fig 2.1 System Architecture

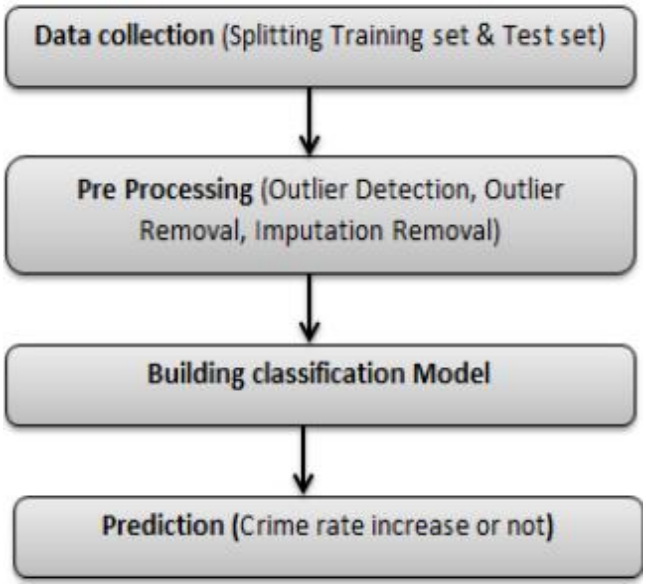


Fig 2.2 Flow Chart

2.3.5 Advantages

- Accuracy of crimes rates decreases.
- Safety and security of people around us. Thus, machine learning algorithms also helps in safety and security system.

2.4 System requirements

Software and Hardware Requirements:

2.4.1 Hardware:

- OS – Windows 7, 8 and 10 (32 and 64 bit)
- RAM – 4GB

2.4.2 Software:

- Python / Anaconda Navigator

2.5 MODULES

2.5.1 DATA COLLECTION

2.5.2 DATA PRE-PROCESSING

2.5.3 FEATURE EXTRATION

2.5.4 EVALUATION MODEL

2.5.1 DATA COLLECTION

Data collection is a process in which information is gathered from many sources which is later used to develop the machine learning models. The data should be stored in a way that makes sense for problem. In this step the data set is converted into the understandable format which can be fed into machine learning models.

Data used in this paper is a set of crime head and year wise total cases occurred records. This step is concerned with selecting the subset of all available data that we will be working with. ML problems start with data preferably, lots of data (examples or observations) for which we already know the target answer. Data for which we already know the target answer is called labelled data.

2.5.2 DATA PRE-PROCESSING

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

Three common data pre-processing steps are:

- Formatting: The data we have selected may not be in a format that is suitable for we to work with. The data may be in a relational database and we would

like it in a flat file, or the data may be in a proprietary file format and we 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 we believe we 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 we need to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. We 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.

2.5.3 FEATURE EXTRATION

Next thing is to do Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Finally, our models are trained using Classifier algorithm. We use classify module on Natural Language Toolkit library on Python. We use the labelled dataset gathered. The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre-processed data.

2.5.4 EVALUATION MODEL

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and over fitted models. There are two methods of evaluating models in data science, Hold-Out and Cross-Validation. Performance of each classification model is estimated base on its averaged. The result will be in the visualized form. Representation of classified data in the form of graphs.

Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

Steps

1. First, we take crime dataset.
2. Filter dataset according to requirements and create a new dataset which has attribute according to analysis to be done.
3. Perform k means clustering on resultant dataset formed.
4. From result plot data between crimes and get required cluster.
5. Analysis can be done on cluster formed by perform Supervised classification algorithm on resultant dataset formed.
6. Finally, we will get results as accuracy metrics.

CHAPTER-3

SUPERVISED ALGORITHM

3.1 Support Vector Machine

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems.

In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features we have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper- plane that differentiates the two classes very well.

Support Vectors are simply the co-ordinates of individual observation. The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line). We can look at support vector machines and a few examples of its working here.

3.2 DATAFLOW DIAGRAM

3.2.1 LEVEL 0

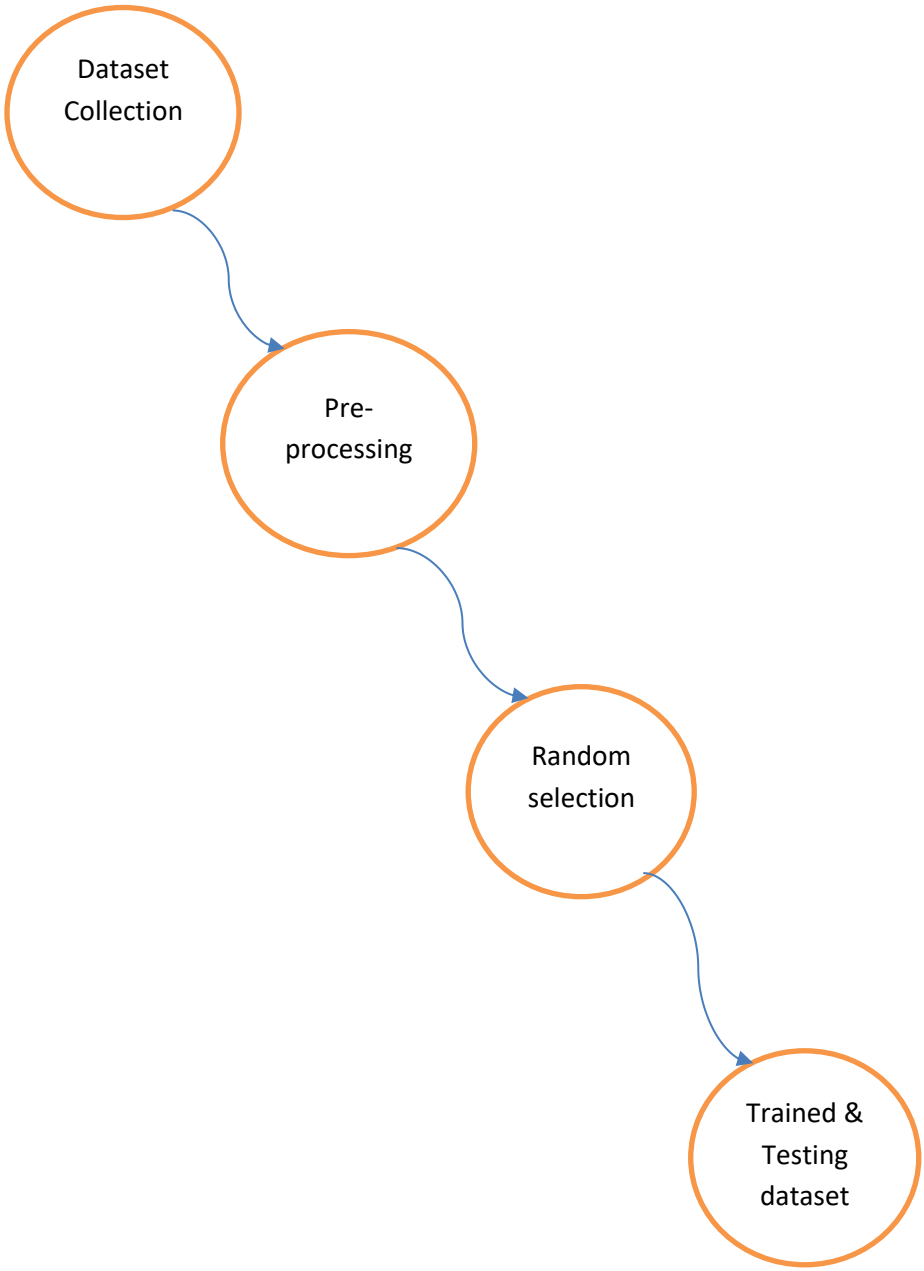


Fig 3.1 Dataflow Diagram

3.2.2 LEVEL 1

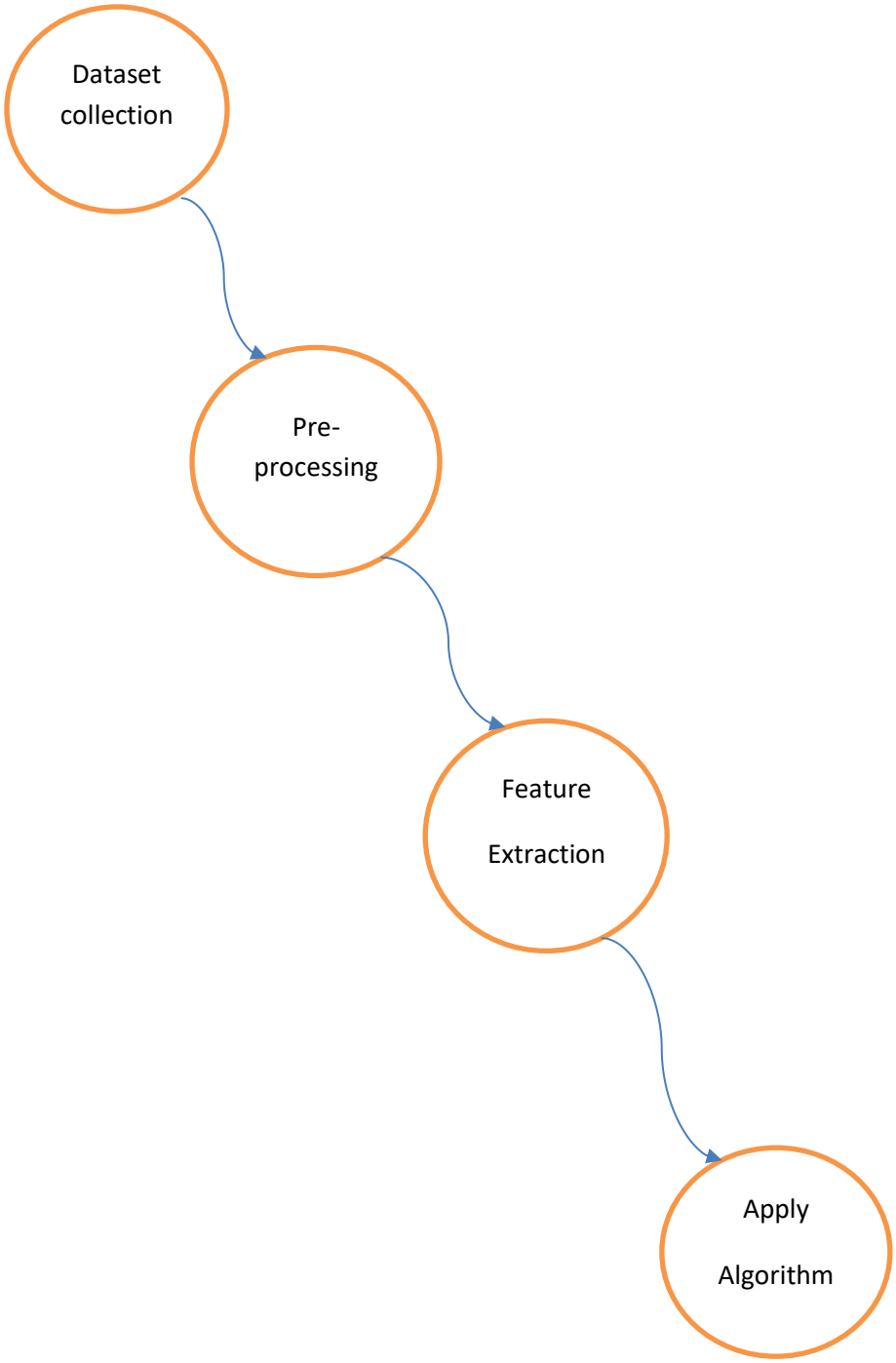


Fig 3.2 Dataflow Diagram

3.2.3 LEVEL 2

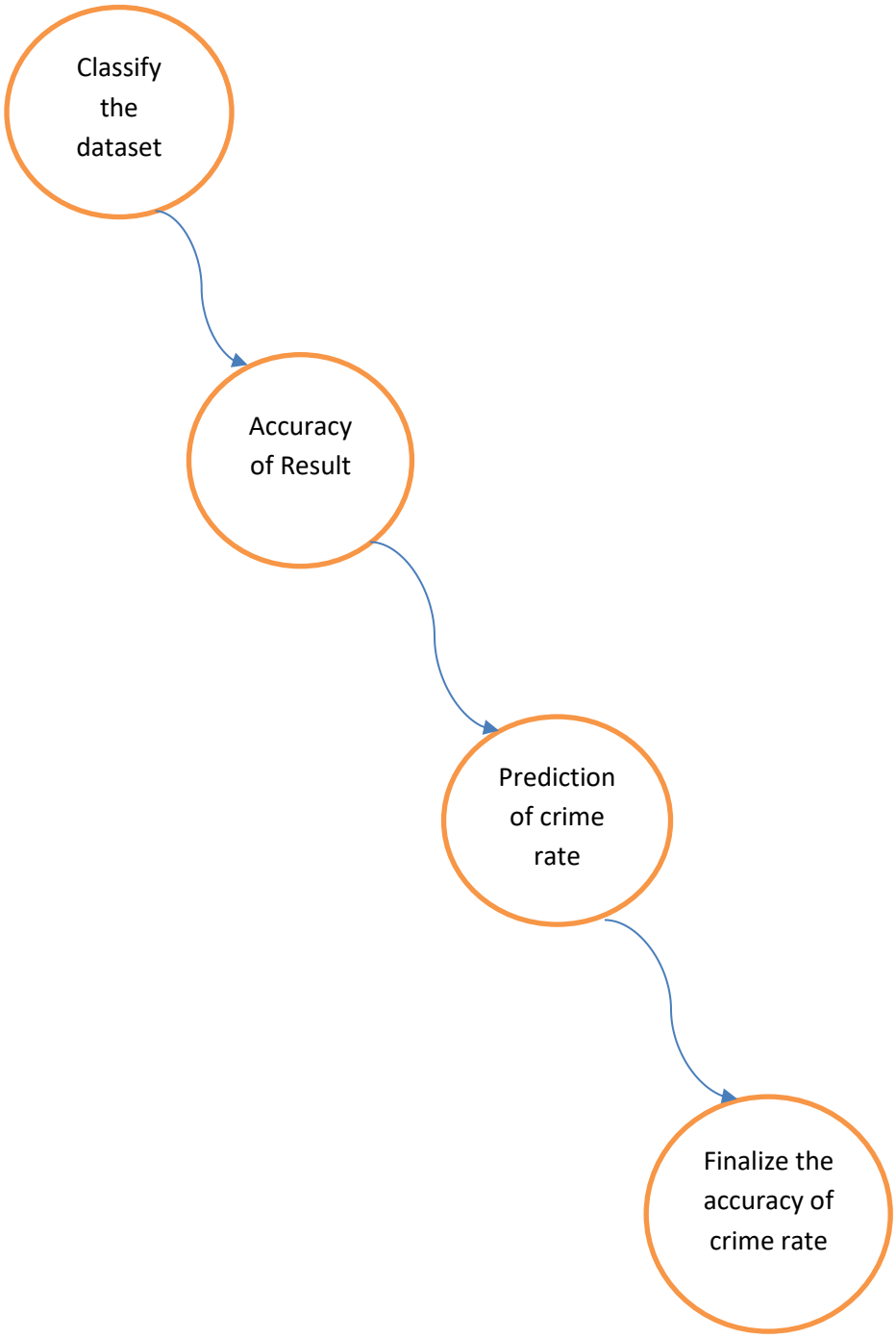


Fig 3.3 Dataflow Diagram

CHAPTER-4

UML DIAGRAMS

4.1 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.

4.2 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 **U**nified **M**odeling **L**anguage.

- UML is a pictorial language used to make software blue prints.

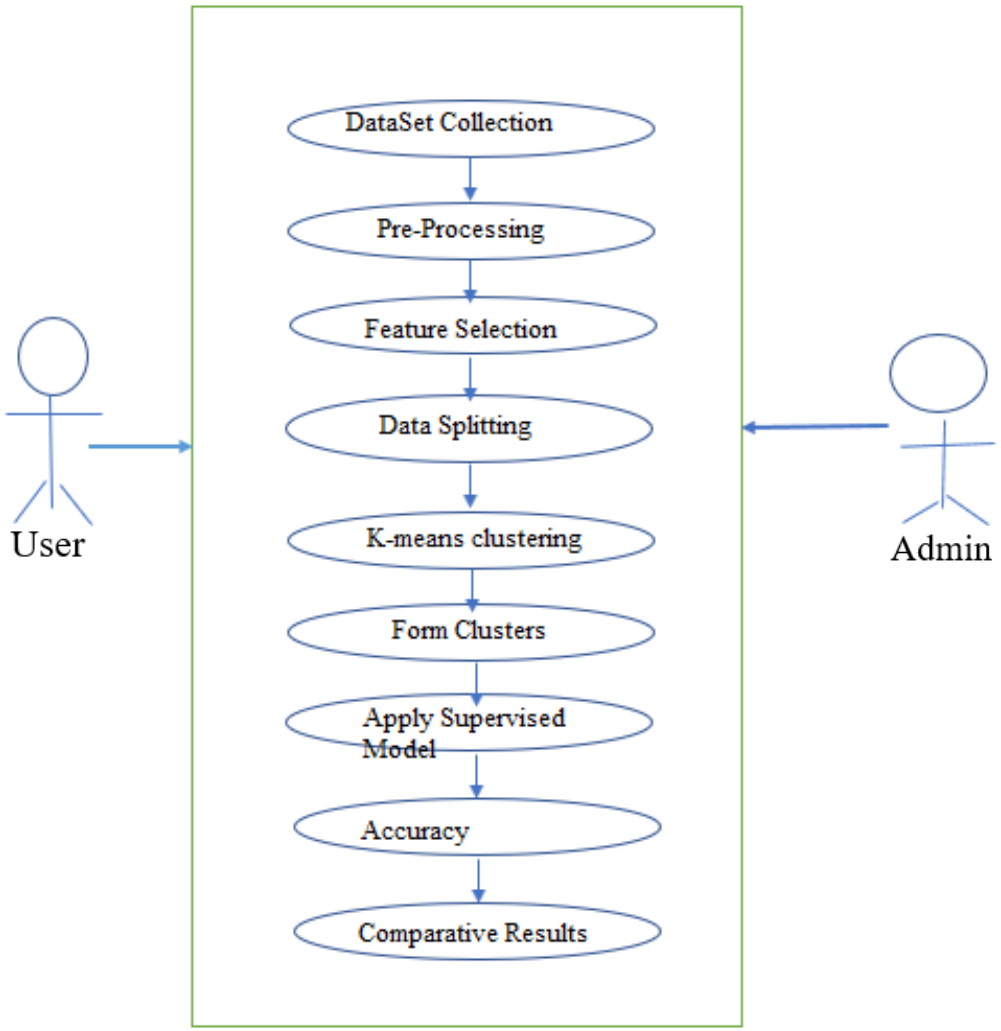


Fig 4.1 Use Case diagram

4.3 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. 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.

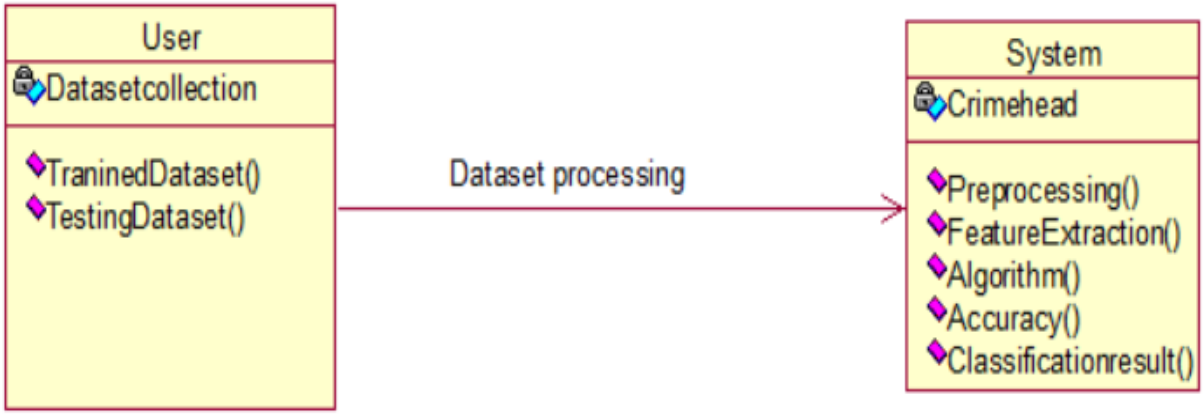


Fig 4.2 Class diagram

4.4 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.

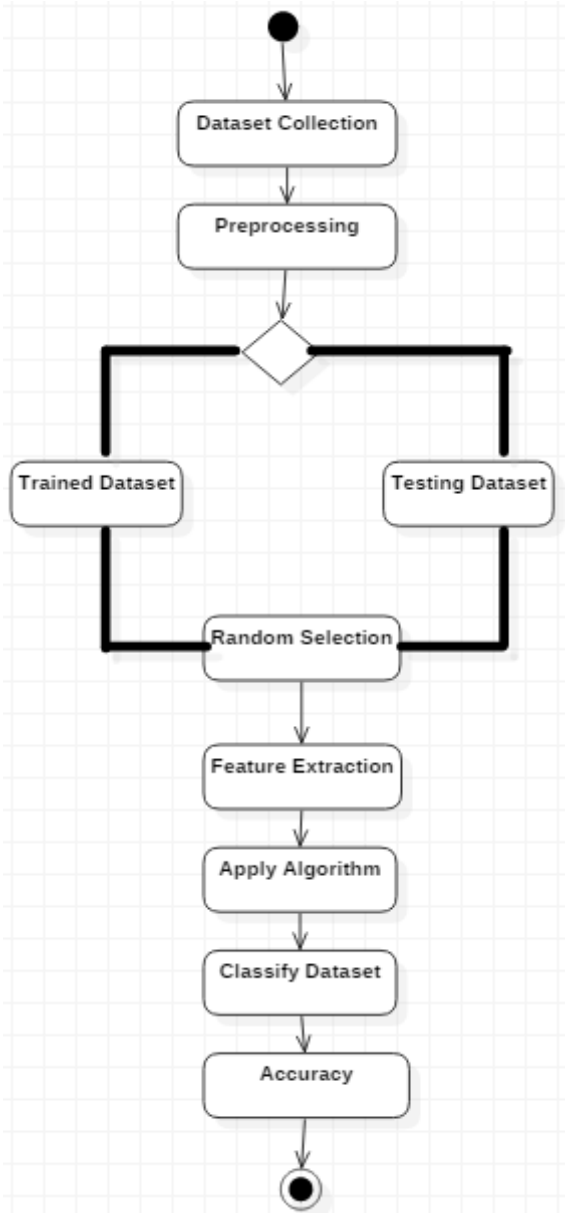


Fig 4.3 Activity Diagram

4.5 Sequence Diagram

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.

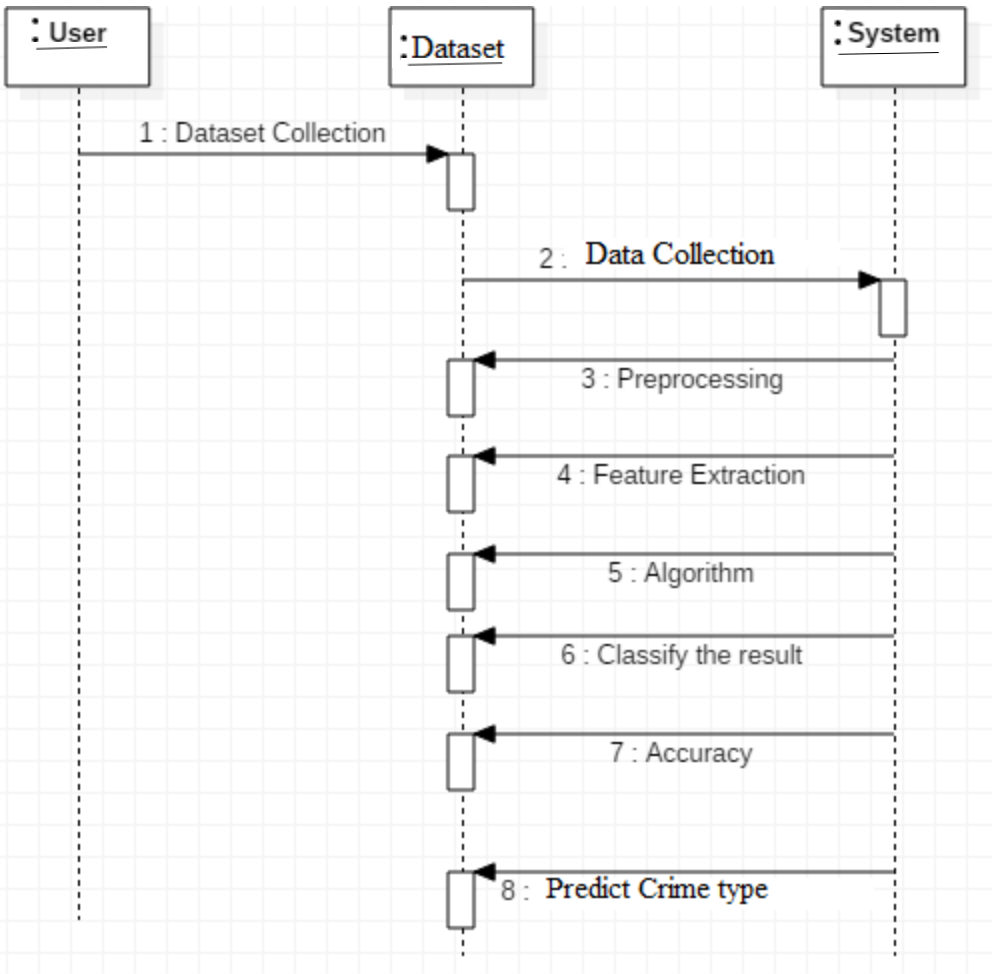


Fig 4.4 Sequence Diagram

CHAPTER-5

DOMAIN SPECIFICATION

5.1 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 answer.

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.

5.2 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.

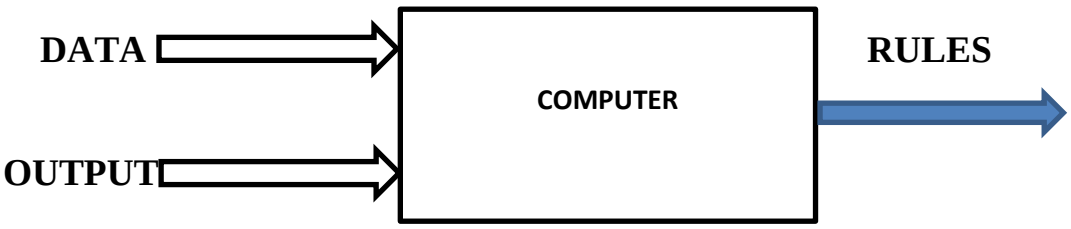


Fig 5.1 Machine Learning

5.3 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 it's 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.** we 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.

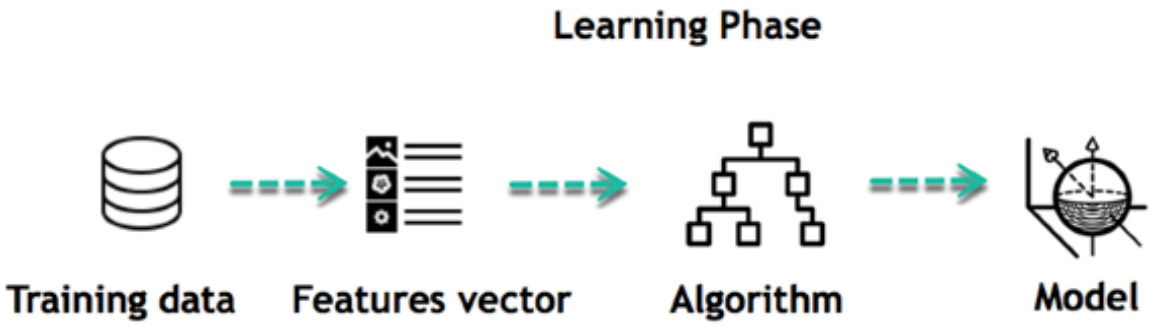


Fig 5.2 Learning Phase

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. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant.

5.4 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. We can use the model previously trained to make inference on new data.

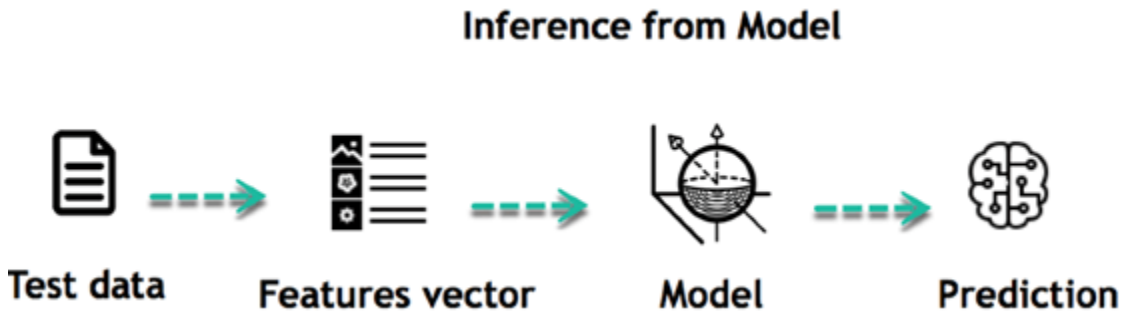


Fig 5.3 Inference from Model

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.

5.4.1 Machine learning Algorithms and where they are used?

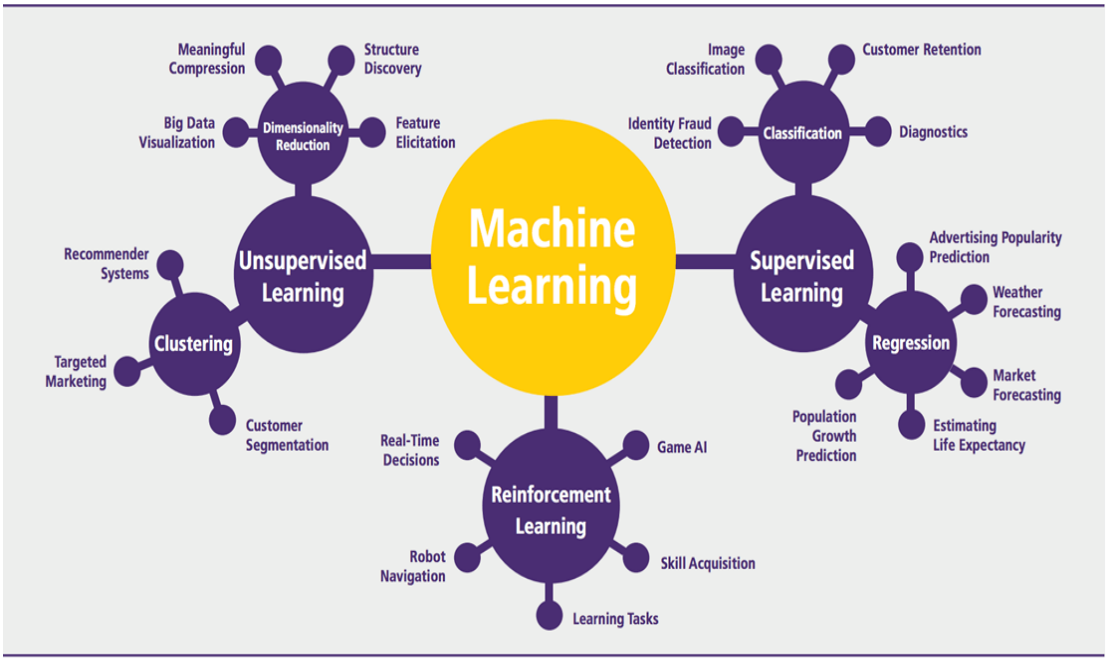


Fig 5.4 Usage of Machine learning Algorithms

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

5.5 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.

We can use supervised learning when the output data is known. The algorithm will predict new data.

There are two categories of supervised learning:

5.5.1 Classification task

5.5.2 Regression task

Algorithm Name	**Description**	**Type**
Linear regression	Finds a way to correlate each feature to the output to help predict future values.	Regression
Logistic regression	Extension of linear regression that's used for classification tasks. The output variable 3is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colors)	Classification
Decision tree	Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes (e.g., if a feature is a color, each possible color becomes a new branch) until a final decision output is made	Regression Classification

Naive Bayes	The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event.	Regression Classification
Support vector machine	Support Vector Machine, or SVM, is typically used for the classification task. SVM algorithm finds a hyperplane that optimally divided the classes. It is best used with a non-linear solver.	Regression (not very common) Classification
AdaBoost	Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome	Regression Classification
Gradient-boosting trees	Gradient-boosting trees is a state-of-the-art classification/regression technique. It is focusing on the error committed by the previous trees and tries to correct it.	Regression Classification

Table 5.1 Supervised learning

5.5.1 Classification

Imagine we want to predict the gender of a customer for a commercial. We will start gathering data on the height, weight, job, salary, purchasing basket, etc. from our customer database. We know the gender of each of our 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 we have collected). When the model learned how to recognize male or female, we can use new data to make a prediction. For instance, we just got new information from an unknown customer, and we 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).

5.5.2 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.

5.6 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)

We can use it when we do not know how to classify the data, and we want the algorithm to find patterns and classify the data for us.

Algorithm	**Description**	**Type**
K-means clustering	Puts data into some groups (k) that each contains data with similar characteristics (as determined by the model, not in advance by humans)	Clustering
Gaussian mixture model	A generalization of k-means clustering that provides more flexibility in the size and shape of groups (clusters	Clustering
Hierarchical clustering	Splits clusters along a hierarchical tree to form a classification system. Can be used for Cluster loyalty-card customer	Clustering
Recommender system	Help to define the relevant data for making a recommendation.	Clustering
PCA/T-SNE	Mostly used to decrease the dimensionality of the data. The algorithms reduce the number of features to 3 or 4 vectors with the highest variances.	Dimension Reduction

Table 5.2 Unsupervised learning

5.7 Application of Machine learning

5.7.1 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.

5.7.2 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.

5.7.3 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.

5.7.4 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.

5.7.5 Healthcare industry

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

5.7.6 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.

5.8 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.

5.9 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.

5.10 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.

5.11 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)

5.12 Applications/ Examples of deep learning applications

5.12.1 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.

5.12.2 AI in HR

Under Armor, a sportswear company revolutionizes hiring and modernizes the candidate experience with the help of AI. In fact, Under Armor Reduces hiring time for its retail stores by 35%. Under Armor 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 Armor's ability to have their retail stores fully staffed, ramped and ready to operate.

At that time, Under Armor 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 armor 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.

5.12.3 AI in Marketing

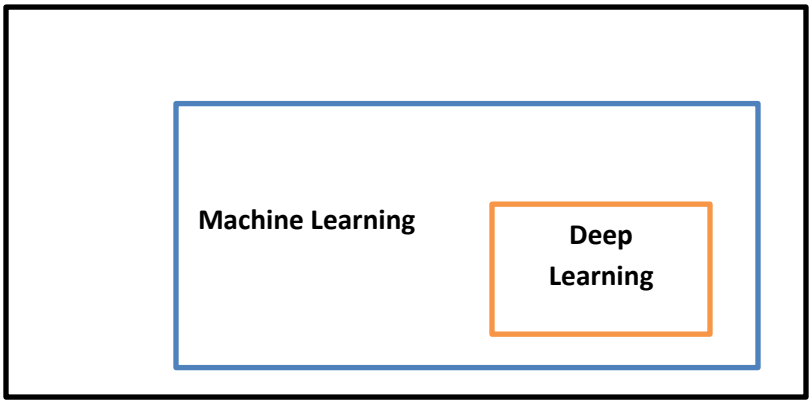
AI is a valuable tool for customer service management and personalization challenges. Improved speech recognition in call-center management and call routing as a result of the application of AI techniques allows a more seamless experience for customers.

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.

	Machine Learning	**Deep Learning**
Data Dependencies	Excellent performances on a small/medium dataset	Excellent performance on a big dataset
Hardware dependencies	Work on a low-end machine.	Requires powerful machine, preferably with GPU: DL performs a significant amount of matrix multiplication
Feature engineering	Need to understand the features that represent the data	No need to understand the best feature that represents the data
Execution time	From few minutes to hours	Up to weeks. Neural Network needs to compute a significant number of weights
Interpretability	Some algorithms are easy to interpret (logistic, decision tree), some are almost impossible (SVM, XGBoost)	Difficult to impossible

Table 5.3 Difference between Machine Learning and Deep Learning



5.13 When to use ML or DL

In the table below, we summarize the difference between machine learning and deep learning.

	Machine learning	**Deep learning**
Training dataset	Small	Large
Choose features	Yes	No
Number of algorithms	Many	Few
Training time	Short	Long

Table 5.4: ML vs DL

With machine learning, we need fewer data to train the algorithm than deep learning. Deep learning requires an extensive and diverse set of data to identify the underlying structure. Besides, machine learning provides a faster-trained model. Most advanced deep learning architecture can take days to a week to train. The advantage of deep learning over machine learning is it is highly accurate. You do not need to understand what features are the best representation of the data; the neural network learned how to select critical features. In machine learning, we need to choose for ourself what features to include in the model.

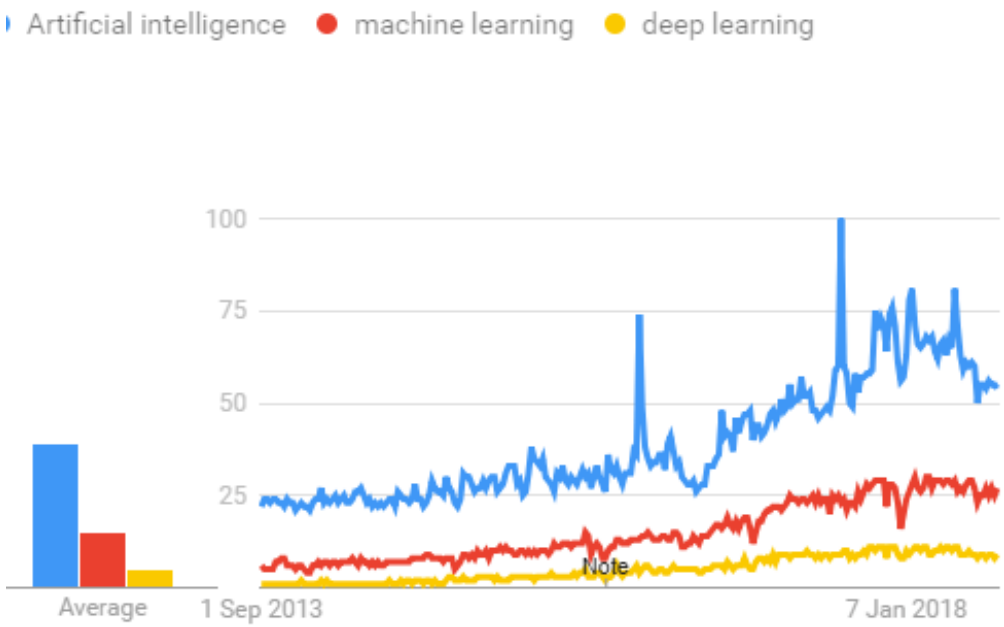


Fig 5.5: AI, ML, DL

CHAPTER-6

TENSORFLOW

6.1 TensorFlow

The most famous deep learning library in the world is Google's TensorFlow. Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations.

To give a concrete example, Google users can experience a faster and more refined the search with AI. If the user types a keyword the search bar, Google provides a recommendation about what could be the next word.

Google wants to use machine learning to take advantage of their massive datasets to give users the best experience. Three different groups use machine learning:

- Researchers
- Data scientists
- Programmers.

They can all use the same toolset to collaborate with each other and improve their efficiency.

Google does not just have any data; they have the world's most massive computer, so TensorFlow was built to scale. TensorFlow is a library developed by the Google Brain Team to accelerate machine learning and deep neural network research.

It was built to run on multiple CPUs or GPUs and even mobile operating systems, and it has several wrappers in several languages like Python, C++ or Java.

6.2 TensorFlow Architecture

Tensor flow architecture works in three parts:

- Preprocessing 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**. We can construct a sort of **flowchart** of operations (called a Graph) that we 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.

6.3 Where can Tensor flow run?

TensorFlow can hardware, and software requirements can be classified into

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

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

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

We can train it on multiple machines then we can run it on a different machine, once we 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.

6.4 List of Prominent Algorithms supported by TensorFlow

- Linear regression: tf.estimator.LinearRegressor
- Classification: tf.Estimator.LinearClassifier
- Deep learning classification: tf.estimator.DNNClassifier
- Booster tree regression: tf.estimator.BoostedTreesRegressor
- Boosted tree classification: tf.estimator.BoostedTreesClassifier

6.5 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.

6.5.1 Python is Interpreted

Python is processed at runtime by the interpreter. We do not need to compile our program before executing it. This is similar to PERL and PHP.

6.5.2 Python is Interactive

We can actually sit at a Python prompt and interact with the interpreter directly to write our programs.

6.5.3 Python is Object-Oriented

Python supports Object-Oriented style or technique of programming that encapsulates code within objects.

6.5.4 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.

6.6 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.

6.7 Python Features

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

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

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

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

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

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

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

viii. **Databases:** Python provides interfaces to all major commercial databases.

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

x. **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.

CHAPTER-7

IMPLEMENTATION

7.1 Anaconda Navigator

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows we 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.

7.2 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. We can use it to find the packages we want, install them in an environment, run the packages and update them, all inside Navigator.

7.3 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 our own Navigator applications

7.4 How can I run code with Navigator?

The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and execute our code.

We can also use Jupyter Notebooks the same way. Jupyter Notebooks are an increasingly popular system that combine our code, descriptive text, output, images and interactive interfaces into a single notebook file that is edited, viewed and used in a web browser.

7.5 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.

7.6 Implementation

```
import matplotlib.pyplot as plt

%matplotlib inline

import numpy as np

import pandas as pd

df = pd.read_csv('2001-2012.csv')

df.head()

years_title = [str(i) for i in range(2001,2013)]

STATES_IN_INDIA = df['STATE/UT'].unique()

STATES_IN_INDIA = STATES_IN_INDIA[:-4]

STATES_IN_INDIA



TYPES_OF_CASES = df['CRIME HEAD'].unique()

TYPES_OF_CASES = TYPES_OF_CASES[:-1]

TYPES_OF_CASES
```

```
for state in STATES_IN_INDIA:
    fig=plt.figure(figsize=(12, 8), dpi= 80, facecolor='w', edgecolor='k')
    plt.title(state)
    plt.xlabel('Years')
    plt.ylabel('No of Cases')
    for case in TYPES_OF_CASES:
        temp_df = df[(df['STATE/UT'] == state ) & (df['CRIME HEAD'] == case)]
        N_cases = [temp_df[c].values[0] for c in years_title]
        plt.plot(years_title,N_cases)
        plt.legend(TYPES_OF_CASES)


fig=plt.figure(figsize=(20, 10), dpi= 80, facecolor='w', edgecolor='k')
plt.title('TOTAL CRIME YEAR WISE')
plt.xlabel('Years')
plt.ylabel('No of Cases')
```

```
for state in STATES_IN_INDIA:

    temp_df = df[(df['STATE/UT'] == state ) & (df['CRIME HEAD'] == 'TOTAL
CRIMES AGAINST WOMEN')]

    N_cases = [temp_df[c].values[0] for c in years_title]

    plt.plot(years_title,N_cases)

    plt.legend(STATES_IN_INDIA)



print('Data set:')

for col_name in df.columns:

    if df[col_name].dtypes == 'object' :

        unique_cat = len(df[col_name].unique())

        print("Feature '{col_name}' has {unique_cat}
categories".format(col_name=col_name, unique_cat=unique_cat))

print()

from sklearn import preprocessing

lab=preprocessing.LabelEncoder()

#df['STATE/UT']=lab.fit_transform(df['STATE/UT'])
```

```
df['CRIME HEAD']=lab.fit_transform(df['CRIME HEAD'])

df.head()



from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters=9)

kmeans.fit(df.iloc[:,1:])



kmeans.cluster_centers_



labels=kmeans.labels_

labels



import numpy as np

unique, counts = np.unique(kmeans.labels_, return_counts=True)

dict_data = dict(zip(unique, counts))

dict_data
```

```
df["cluster"] = kmeans.labels_



import seaborn as sns

sns.lmplot('2011', '2012', data=df, hue='cluster', palette='coolwarm', size=5,

aspect=1, fit_reg=False)



# Inertia is the sum of squared error for each cluster.

# Therefore the smaller the inertia the denser the cluster(closer together all the

points are)

kmeans.inertia_



kmeans.score



cust = [[7,871,1002,946,1016,935,1049,1070,1257,1188,1362,1442,1341]]

kmeans.predict(cust)[0]

# Initialize the matplotlib figure

f, ax = plt.subplots(figsize=(24, 15))
```

```
# Load the dataset

stats = df.sort_values([ "cluster", "STATE/UT"], ascending=True)

sns.set_color_codes("pastel")

sns.barplot(y="STATE/UT", x="2012", data=stats)

sns.despine(left=True, bottom=True)



X = df.iloc[:,1:14]

y = df.iloc[:,df.columns=='cluster']

print(X.head())

y.head()



from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,y, test_size = 0.2,

random_state=0)

from sklearn import svm

sv=svm.LinearSVC()

sv.fit(X_train,y_train)

predic3=sv.predict(X_test)
```

```
from sklearn.metrics import accuracy_score,classification_report

acc3=accuracy_score(predic3,y_test)

print(acc3)

clf3=classification_report(predic3,y_test)

print(clf3)



from sklearn.cluster import KMeans

km=KMeans()

km.fit(X_train,y_train)

predic3=km.predict(X_test)

acc4=accuracy_score(predic3,y_test)

print(acc4)

clf4=classification_report(predic3,y_test)

print(clf4)



import matplotlib.pyplot as plt; plt.rcdefaults()

objects = ('Support Vector','K Means')
```

```
y_pos = np.arange(len(objects))

performance = [acc3,acc4]

plt.bar(y_pos, performance, align='center', alpha=0.5)

plt.xticks(y_pos, objects)

plt.ylabel('Accuracy')

plt.title('SVM vs KMeans')

plt.show()



df.T
```

Table 7.1 Python Code

CHAPTER-8

TESTING

8.1 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.

8.2 TESTING METHODS

8.2.1 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

8.2.2 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.

Test Case for Excel Sheet Verification:

Here in machine learning we are dealing with dataset which is in excel sheet format so if any test case we need means we need to check excel file. Later on, classification will work on the respective columns of dataset.

Test Case:

SL #	TEST CASE NAME	DESCRIPTION	STEP NO	ACTION TO BE TAKEN (DESIGN STEPS)	EXPECTED (DESIGN STEP)	Test Execution Result (PASS/FAIL)
1	Excel Sheet verification	**Objective:** There should be an excel sheet. Any number of rows can be added to the sheet.	Step 1	Excel sheet should be available	Excel sheet is available	Pass
			Step 2	Excel sheet is created based on the template	The excel sheet should always be based on the template	Pass
			Step 3	Changed the name of excel sheet	Should not make any modification on the name of excel sheet	Fail
			Step 4	Added 10000 or above records	Can add any number of records	Pass

Table 8.1 Test Case

CHAPTER-9

RESULTS

9.1 RESULTS

Data mining is a process to extract knowledge from existing data. It is used as a tool in banking and finance, in general, to discover useful information from the operational and historical data to enable better decision-making. It is an interdisciplinary field, the confluence of Statistics, Database technology, Information science, Machine learning, and Visualization. It involves steps that include data selection, data integration, data transformation, data mining, pattern evaluation, knowledge presentation.

9.2 Visualization

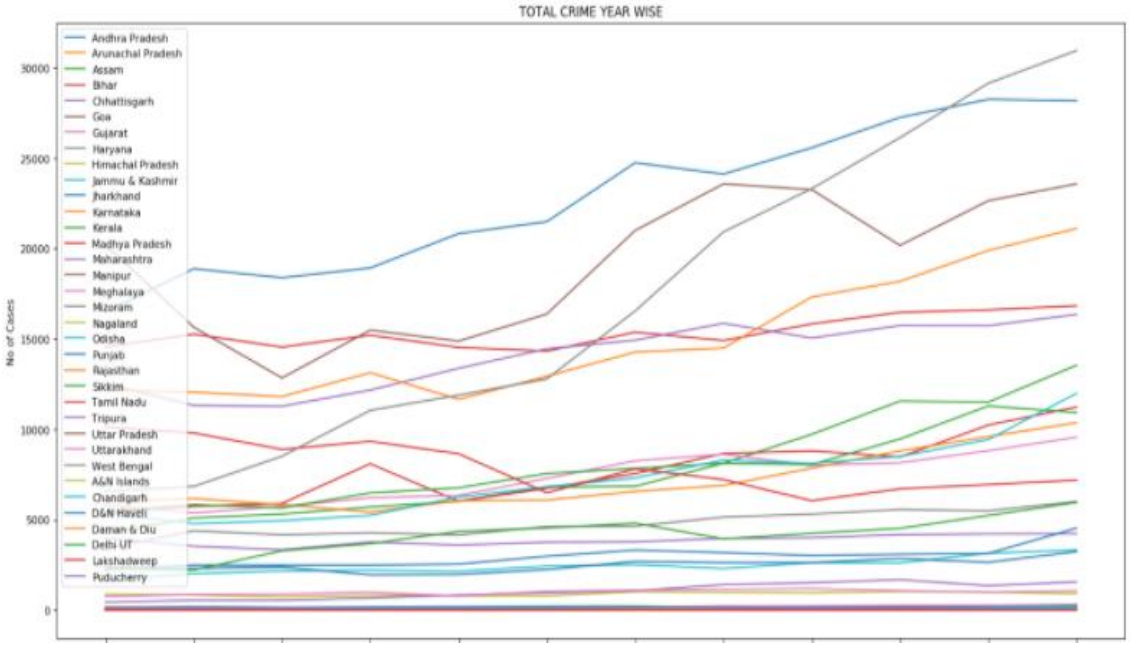


Fig 9.1 Total Crimes Year Wise Graph

```
array([[3.84810127e+00, 1.08202532e+02, 1.16759494e+02, 1.08848101e+02,
        1.15059072e+02, 1.16232068e+02, 1.24852321e+02, 1.31649789e+02,
        1.32523207e+02, 1.30611814e+02, 1.38599156e+02, 1.42139241e+02,
        1.53265823e+02],
       [8.00000000e+00, 1.43795000e+05, 1.43034000e+05, 1.40601000e+05,
        1.54333000e+05, 1.55553000e+05, 1.64765000e+05, 1.85312000e+05,
        1.95857000e+05, 2.03804000e+05, 2.13585000e+05, 2.28650000e+05,
        2.44270000e+05],
       [7.40000000e+00, 1.47988000e+04, 1.44496000e+04, 1.37746000e+04,
        1.58528000e+04, 1.63380000e+04, 1.74812000e+04, 2.06856000e+04,
        2.25996000e+04, 2.38536000e+04, 2.51010000e+04, 2.79578000e+04,
        2.91734000e+04],
       [1.00000000e+00, 4.91700000e+04, 4.92370000e+04, 5.07030000e+04,
        5.81210000e+04, 5.83190000e+04, 6.31280000e+04, 7.59300000e+04,
        8.13440000e+04, 8.95460000e+04, 9.40410000e+04, 9.91350000e+04,
        1.06527000e+05],
       [4.78571429e+00, 6.48300000e+03, 6.47285714e+03, 6.40192857e+03,
        6.85435714e+03, 6.76664286e+03, 7.12450000e+03, 8.03564286e+03,
        8.38014286e+03, 8.58707143e+03, 8.94914286e+03, 9.39342857e+03,
        1.00462857e+04],
       [0.00000000e+00, 3.41240000e+04, 3.39430000e+04, 3.29390000e+04,
        3.45670000e+04, 3.41750000e+04, 3.66170000e+04, 3.87340000e+04,
        4.04130000e+04, 3.87110000e+04, 4.06130000e+04, 4.29680000e+04,
        4.53510000e+04],
       [4.07142857e+00, 3.22171429e+03, 3.13521429e+03, 3.10414286e+03,
        3.33350000e+03, 3.40278571e+03, 3.60964286e+03, 3.83071429e+03,
        3.93257143e+03, 4.06378571e+03, 4.34750000e+03, 4.58221429e+03,
        4.95407143e+03],
       [4.02127660e+00, 1.33634043e+03, 1.34053191e+03, 1.32693617e+03,
        1.42453191e+03, 1.47565957e+03, 1.46836170e+03, 1.58897872e+03,
        1.67787234e+03, 1.70470213e+03, 1.76544681e+03, 1.90057447e+03,
        2.05076596e+03],
       [6.25000000e+00, 1.07767500e+04, 1.06750000e+04, 1.06450000e+04,
        1.17082500e+04, 1.16230000e+04, 1.22802500e+04, 1.36160000e+04,
        1.47310000e+04, 1.60757500e+04, 1.70457500e+04, 1.79967500e+04,
        1.85390000e+04]])
```

Table 9.1 Clusters

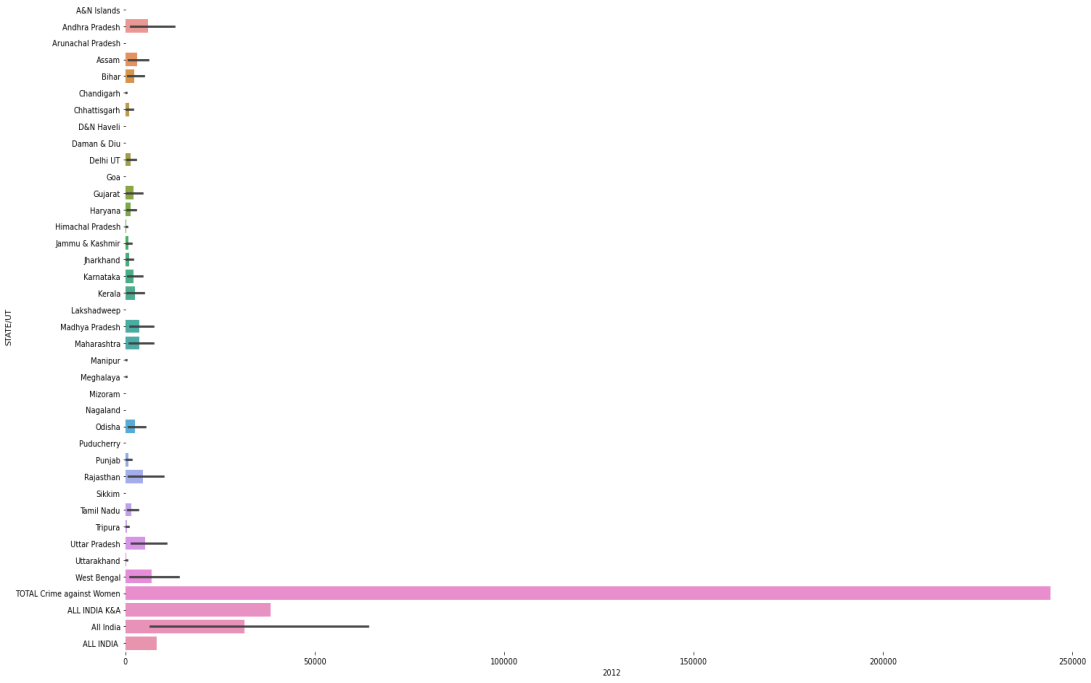


Fig 9.2 Graph for Crime rate in 2012

```
from sklearn import svm

sv=svm.LinearSVC()

sv.fit(X_train,y_train)

predic3=sv.predict(X_test)
from sklearn.metrics import accuracy_score,classification_report
acc3=accuracy_score(predic3,y_test)
print(acc3)

clf3=classification_report(predic3,y_test)
print(clf3)
```

```
0.6923076923076923
              precision    recall  f1-score   support

           0       0.87      0.98      0.92        46
           1       0.00      0.00      0.00         9
           2       0.00      0.00      0.00         1
           3       0.00      0.00      0.00         0
           4       0.00      0.00      0.00         0
           5       0.00      0.00      0.00         3
           6       0.00      0.00      0.00         0
           7       0.00      0.00      0.00         0
           8       0.00      0.00      0.00         6

    accuracy                           0.69        65
   macro avg       0.10      0.11      0.10        65
weighted avg       0.61      0.69      0.65        65
```

Fig 9.3 Accuracy result with SVM

```
from sklearn.cluster import KMeans
km=KMeans()
km.fit(X_train,y_train)

predic3=km.predict(X_test)

acc4=accuracy_score(predic3,y_test)
print(acc4)

clf4=classification_report(predic3,y_test)
print(clf4)
```

```
0.8923076923076924
              precision    recall  f1-score   support

           0       1.00      0.93      0.96        56
           3       1.00      1.00      1.00         1
           4       1.00      1.00      1.00         2
           6       1.00      0.50      0.67         6
           7       0.00      0.00      0.00         0

    accuracy                           0.89        65
   macro avg       0.80      0.69      0.73        65
weighted avg       1.00      0.89      0.94        65
```

Fig 9.4 Accuracy result with K-Means

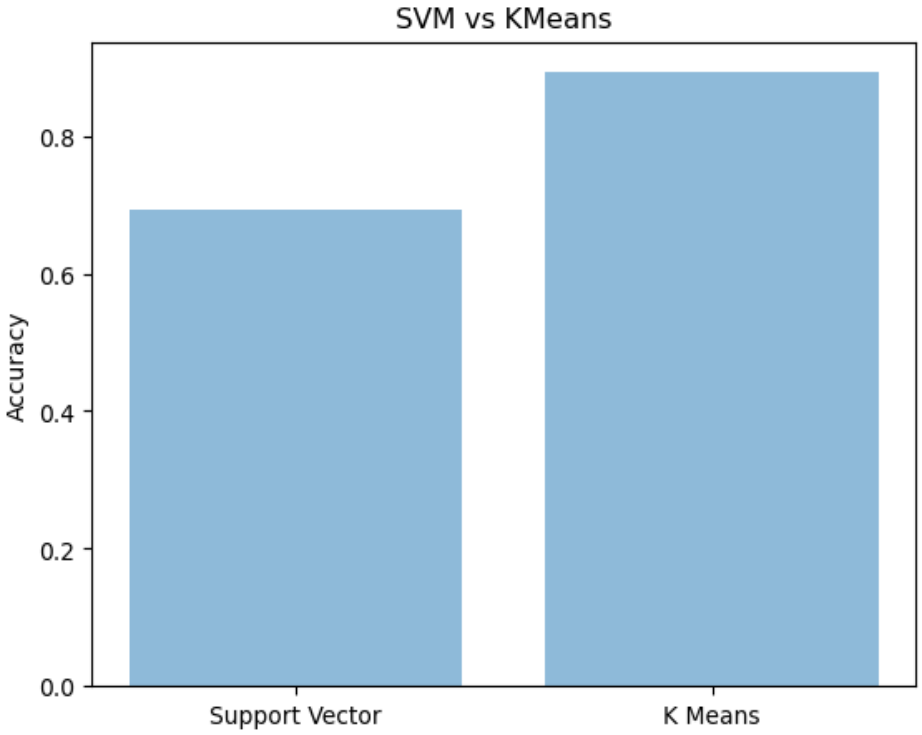


Fig 9.5 Comparative model Accuracy graph

9.3 Predicted Crime rate

	0	1	2	3	4	5	6	7	8	9	...	314	315	316	317	318	319
STATE/UT	Andhra Pradesh	Arunachal Pradesh	Assam	Bihar	Chhattisgarh	Goa	Gujarat	Haryana	Himachal Pradesh	Jammu & Kashmir	...	Uttarakhand	West Bengal	A&N Islands	Chandigarh	D&N Haveli	Daman & Diu
CRIME HEAD	7	7	7	7	7	7	7	7	7	7	...	8	8	8	8	8	8
2001	871	33	817	888	959	12	286	398	124	169	...	749	6570	34	150	19	10
2002	1002	38	970	1040	992	12	267	361	137	192	...	870	6842	27	189	15	8
2003	946	31	1095	985	898	31	236	353	126	211	...	886	8508	22	159	13	10
2004	1016	42	1171	1390	969	37	339	386	153	218	...	988	11047	27	188	22	7
2005	935	35	1238	1147	990	20	324	461	141	201	...	786	11887	22	205	24	10
2006	1049	37	1244	1232	995	21	354	608	113	250	...	1038	12785	36	224	32	9
2007	1070	48	1437	1555	982	20	316	488	159	288	...	1097	16544	56	230	18	11
2008	1257	42	1438	1302	978	30	374	631	157	219	...	1151	20912	80	143	28	15
2009	1188	59	1631	929	976	47	433	603	183	237	...	1188	23307	92	150	20	13
2010	1362	47	1721	795	1012	36	408	720	160	245	...	1074	26125	85	141	30	14
2011	1442	42	1700	934	1053	29	439	733	168	277	...	996	29133	51	156	18	11
2012	1341	46	1716	927	1034	55	473	668	183	303	...	1067	30942	49	241	16	11
cluster	7	0	7	7	7	0	0	0	0	0	...	7	2	0	0	0	0

15 rows × 324 columns

Table 9.2 Predicted Crime rate

CHAPTER-10

CONCLUSION

Crime prediction is one the current trends in the society. Crime prediction intends to reduce crime occurrences. It does this by predicting which type of crime may occur in future. Here, analysis of crime and prediction are performed with the help of various approaches. From the results obtained we saw that the training time of SVM is very high thus it should be avoided for this dataset.

However which model will work best is totally dependent on the dataset that is being used. In this system, we get to classify and cluster to improve the accuracy of location and pattern-based crimes. This software predicts frequently occurring crimes, especially for particular state, and occurrences.

CHAPTER-11

FUTURE SCOPE

As of now, the project relies on manual input from a human (a police officer) in order to enter details in the database. If we can make this a centralized system and connect it to all the police stations countrywide and make FIR reporting digital, then it would be quite easier to predict crimes in that particular location and recognize patterns in them. It would also encourage citizens to track their E-FIR online. We can also avoid corruption as the government can keep a track on the number of cases registered and their solvability rate which can help them utilize their resources better.

CHAPTER-12

REFERENCES

- Andrey Bogomolov, Bruno Lepri, Jacopo Staiano, Nuria Oliver, Fabio Pianesi, Alex Pentland.” Once Upon a Crime: Towards Crime Prediction from Demographics and Mobile Data”, in ACM International Conference on Multimodal Interaction (ICMI 2014).
- Shiju Sathyadevan, Devan M. S, Surya S Gangadharan, First,” Crime Analysis and Prediction Using Data Mining” International Conference on Networks Soft Computing (ICNSC), 2014.
- Sunil Yadav, Meet Timbadia, Ajit Yadav, Rohit Vishwakarma and Nikhilesh Yadav, Crime pattern detection, analysis and prediction, International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2017.
- Amanpreet Singh,Narina Thakur, Aakanksha Sharma, “A review of supervised machine learning algorithms”, 3rd International Conference on Computing for Sustainable Global Development,2016.
- Bin Li, Yajuan Guo, Yi Wu, Jinming Chen, Yubo Yuan, Xiaoyi Zhang, “An unsupervised learning algorithm for the classification of the protection device in the fault diagnosis system”, in China International Conference on Electricity Distribution (CICED),2014.
- Varshitha D N Vidyashree K P,Aishwarya P Janya T S,K R Dhananjay Gupta Sahana R,”Paper on Different Approaches for Crime Prediction

system",International Journal of Engineering Research Technology (IJERT), ISSN: 2278-0181, 2017

- R. Iqbal, M. A. A. Murad, A. Mustapha, P. H. Shariat Panahy, and N. Khanahmadliravi," An experimental study of classification algorithms for crime prediction", Indian J. of Sci. and Technol., vol. 6, no. 3, pp. 4219- 4225, Mar. 2013.
- Malathi. A, Dr. S. Santhosh Baboo," An Enhanced Algorithm to Predict a Future Crime using Data Mining", International Journal of Computer Applications (0975 – 8887) Volume 21– No.1, May 2011.
- T. Beshah and S. Hill, "Mining Road traffic accident data to improve safety: role of road-related factors on accident severity in Ethiopia", Proc. of Artificial Intell. for Develop. (AID 2010), pp. 14-19, 2010.
- K.B.S. Al-Janabi, "A Proposed Framework for Analyzing Crime Data Set using Decision Tree and Simple K-Means Mining Algorithm," in Journal of Kufa for Mathematics and Computer, Vol. 1, No. 3, 2011, pp. 8-24.
- A. Malathi, S.S. Baboo, "An Enhanced Algorithm to Predict a Future Crime using Data Mining," in International Journal of Computer Applications, Vol. 21, 2011, pp. 1-6.