A Comparative Analysis of Bangla Crime News Categorization Using Most Prominent Machine Learning Algorithms

by

Bristy Dhar
ID: CSE1803015092

Md. Niaj Morshed
ID: CSE1803015063

Supervised by
Salma Tabashum

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)
September 2022

A Comparative Analysis of Bangla Crime News Categorization Using Most Prominent Machine Learning Algorithms

by

Bristy Dhar
ID: CSE1803015092

Md. Niaj Morshed
ID: CSE1803015063

Supervised by
Salma Tabashum

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)
September 2022

APPROVAL

The Thesis titled "**A Comparative Analysis of Bangla Crime News Categorization Using Most Prominent Machine Learning Algorithms**" submitted by Bristy Dhar (CSE1803015092), Md. Niaj Morshed (CSE1803015063) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Supervisor

Salma Tabashum
Lecturer,
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1

(Examiner Name& Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2

(Examiner Name& Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3

(Examiner Name& Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Salma Tabashum,** Lecturer, Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this Thesis has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

(Salma Tabashum)
Supervisor

Signature

Bristy Dhar
ID: CSE1803015092

Niaj Morshed
ID: CSE1803015063

ABSTRACT

This work is dedicated to Bangla Crime Type Classification. Despite several comprehensive crime textual datasets are available on different languages but there is no available dataset on Bengali language. In this work, we created a dataset of Bangla crime articles from different news portals like (Prothom Alo, Jugantor, Noya Digonto), which contains around 3,500 articles. Then we have built our crime classifier model and trained the classifier with our own dataset. This paper explores the use of machine learning approaches, or more specifically, four supervised learning Methods, namely Logistic Regression, Support Vector Machine (SVM), K-Nearest Neighbour (KNN) and Naive Bayes (NB) for categorization of Bangla Crime News Articles. Finally we have summarized the experimental result in tabular form. We can see that significant improved accuracy can be achieved using machine learning algorithms in classifying Bangla Crime data. The final experimental result shows that proposed model is able to achieve around 87% accuracy.

ACKNOWLEDGMENT

At the very beginning, we would like to express our deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the schedule time.

We would then like to thank our supervisor, **Salma Tabashum** for introducing us to the amazingly interesting world of Data Mining Machine Learning and Data Mining. And she is the person who taught us how to perform research work efficiently. Without she and her continuous supervision, guidance and valuable advice, it would have been impossible for us to come at this point and have some output from the thesis. We are especially grateful to Ma'am for allowing us greater freedom in choosing the problems to work on, for his encouragement at times of disappointment.

We would like to convey our gratitude to **Prof. Dr Md Alamgir Hossain**, (Dean, Faculty of Science & Engineering) and special gratitude our honorable departmental head **Bulbul Ahamed**, (Associate Professor & Head, Department of Computer Science and Engineering) for their kind concern, discretion, friendly behavior and precious suggestions.

We would like to express our gratitude to all our teachers. Their motivation and encouragement in addition to the education they provided meant a lot to us.

Last but not the least, we would like to thank our family, our parents, for their encouragement, endless love and for supporting us spiritually throughout our lives.

LIST OF ABBREVIATIONS

BOW	Bag Of Words
KNN	K-Nearest Neighbour
LR	Logistic Regression
ML	Machine Learning
NB	Naive Bayes
NLP	Natural Language Processing
ROI	Return Of Investment
SVM	Support Vector Machine

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Crime data analysis is a systematic analysis for detecting and analyzing various types of crimes and classifies its patterns. These patterns play an important role for solving different crime types, problems and in making different strategies to solve the crime problems. Different types of news articles of online newspapers publish thousands of crime news which contain the details of victims, crime type, criminals, locations etc. Many studies have discovered various techniques to investigate the crime data. Now various machine learning techniques are utilized for the extraction of appropriate meaningful features in order to classify the text documents. In the active research area of text mining, Text categorization is one where all the documents are categorized with basic three types of knowledge, supervised, semi-supervised and unsupervised. Among these machine learning techniques, supervised learning has become more popular in the word. Different types of supervised learning approaches have been used in many research such as Support Vector Machine, Logistic Regression, K-Nearest Neighbour, Naive Bayes, etc. In order to train these supervised learning model, different statistical and machine learning approaches have been used to extract meaningful features for accurate text classifications [1].Despite several comprehensive crime textual datasets are available on different languages but there is no available dataset on Bengali language. In this work, we created a dataset of Bangla crime articles from different news portals like (Prothom Alo, Jugantor, Noya Digonto), which contains around 3,500 articles. Then we have built our crime classifier model and trained the classifier with our own dataset. Finally we have summarized the experimental result in tabular form. We can see that significant improved accuracy can be achieved using machine learning algorithms in classifying Bangla Crime data.

1.2 What is Natural Language Processing?

Natural language processing (NLP) is a field of artificial intelligence in which computers analyze, understand, and derive meaning from human language in a smart and useful way. By utilizing NLP, developers can organize and structure knowledge to perform tasks such as automatic summarization, translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation.

“Apart from common word processor operations that treat text like a mere sequence of symbols, NLP considers the hierarchical structure of language: several words make a phrase, several phrases make a sentence and, ultimately, sentences convey ideas,” John Rehling, an NLP expert at Meltwater Group, says in How Natural Language Processing Helps Uncover Social Media Sentiment. “By analyzing language for its meaning, NLP systems have long filled useful roles, such as correcting grammar, converting speech to text and automatically translating between languages.”

NLP is used to analyze text, allowing machines to understand how humans speak. This human-computer interaction enables real-world applications like automatic text summarization, sentiment analysis, topic extraction, named entity recognition, parts-of-speech tagging, relationship extraction, stemming, and more. NLP is commonly used for text mining, machine translation, and automated question answering.

NLP is characterized as a difficult problem in computer science. Human language is rarely precise, or plainly spoken. To understand human language is to understand not only the words, but the concepts and how they're linked together to create meaning. Despite language being one of the easiest things for the human mind to learn, the ambiguity of language is what makes natural language processing a difficult problem for computers to master [2].

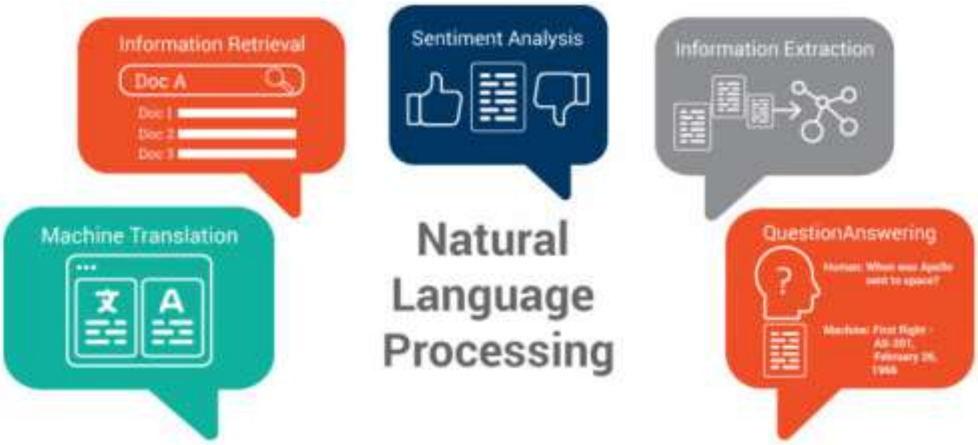


Figure 1.1: Natural Language Processing

Most NLP techniques rely on machine learning to derive meaning from human languages. In fact, a typical interaction between humans and machines using Natural Language Processing could go as follows:

1. A human talks to the machine.
2. The machine captures the audio.
3. Audio to text conversion takes place.
4. Processing of the text's data.
5. Data to audio conversion takes place.
6. The machine responds to the human by playing the audio file.

1.3 Why is NLP important?

The biggest benefit of NLP for businesses is the ability of technology to detect, and process massive volumes of text data across the digital world including; social media platforms, online reviews, news reports, and others. Also, by collecting and analyzing business data, NLP is able to offer businesses valuable insights into brand performance. In addition, NLP models can detect any persisting issues and take necessary mitigation measures to improve performance. Google speech to text is able to achieve all of this by training machines to understand human language in a faster, more accurate, and consistent way than human agents. The technology is able to consistently monitor and process data. This helps brands remain updated with their online presence, and not get riddled with inconsistencies.

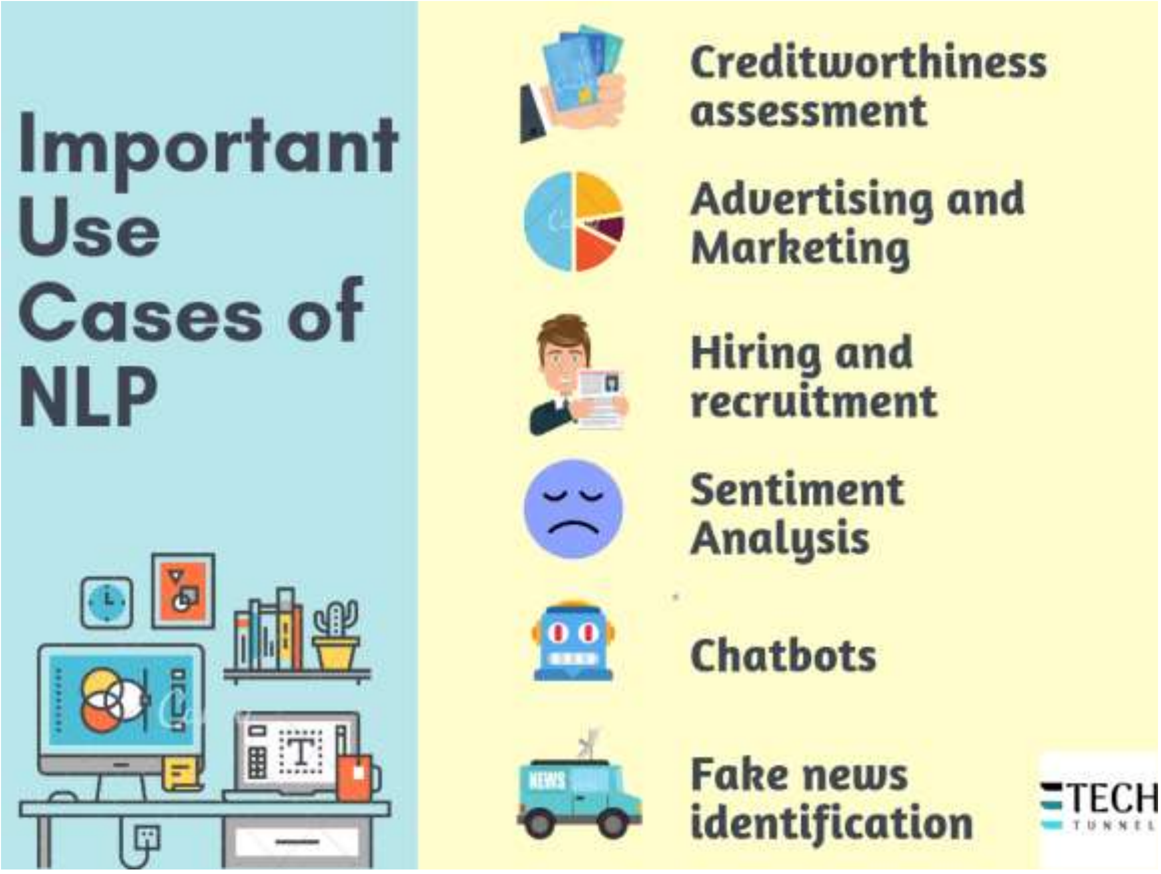
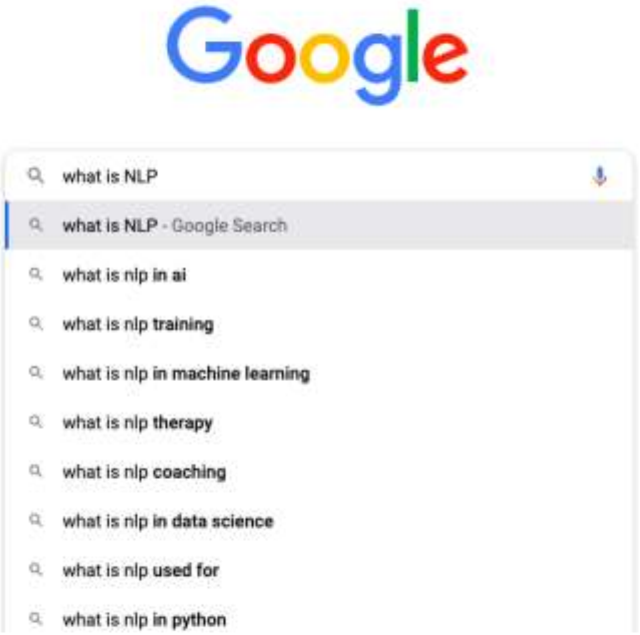


Figure 1.2: Important of Natural Language Processing

1.4 Real Life Applications of NLP

Natural Language Processing is the driving force behind the following common applications:

1.Search Engine Results: An example of NLP in action is search engine functionality. Search engines leverage NLP to suggest relevant results based on previous search history behavior and user intent. When using Google, for example, the search engine predicts what we will continue typing based on popular searches, while also looking at the context and recognizing the meaning behind what we want to say (as opposed to the literal words being typed). It might feel like our thought is being finished before we get the chance to finish typing. This could look like putting in a math equation and having a calculator come up, or even typing a flight number and receiving the flight status breakdown.



what is NLP

what is NLP - Google Search

what is nlp in ai

what is nlp training

what is nlp in machine learning

what is nlp therapy

what is nlp coaching

what is nlp in data science

what is nlp used for

what is nlp in python

Figure 1.3: Search Engine Results

2.Language Translation: One of the most common NLP examples is translation. In the 1950s, Georgetown and IBM presented the first NLP-based translation machine, which had the ability to translate 60 Russian sentences to English automatically. Translation applications available today use NLP and machine learning to accurately translate both text and voice formats for most global languages.

3.Survey Analytics: Data analysis has come a long way in interpreting survey results, although the final challenge is making sense of open-ended responses and unstructured text. NLP, with the support of other AI disciplines, is working towards making these advanced analyses possible. Thanks to NLP, we can analyse our survey responses accurately and effectively without needing to invest human resources in this process. SpaCy and Gensim are examples of code-based libraries that are simplifying the process of drawing insights from raw text.

4.Sentiment Analysis: Oftentimes, when businesses need help understanding their customer needs, they turn to sentiment analysis. Sentiment analysis (also known as opinion mining) is an NLP strategy that can determine whether the meaning behind data is positive, negative, or neutral. For instance, if an unhappy client sends an email which mentions the terms “error” and “not worth the price”, then their opinion would be automatically tagged as one with negative sentiment.

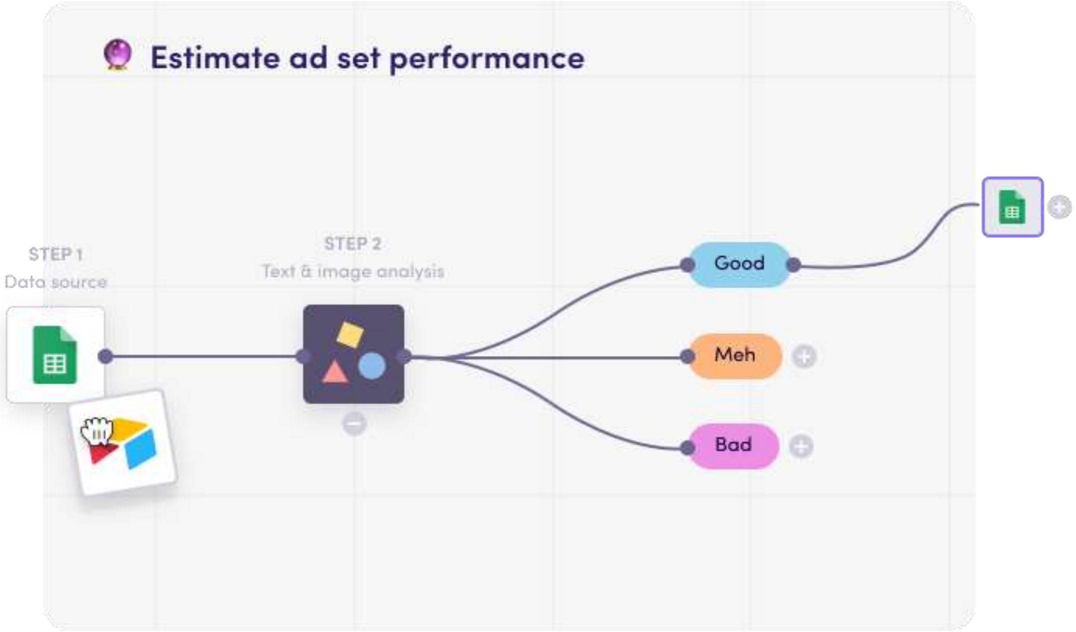


Figure 1.4: Sentiment Analysis workflow in Levity

5. Email filters: Email filters are common NLP examples you can find online across most servers. Spam filters are where it all started – they uncovered patterns of words or phrases that were linked to spam messages. Since then, filters have been continuously upgraded to cover more use cases.

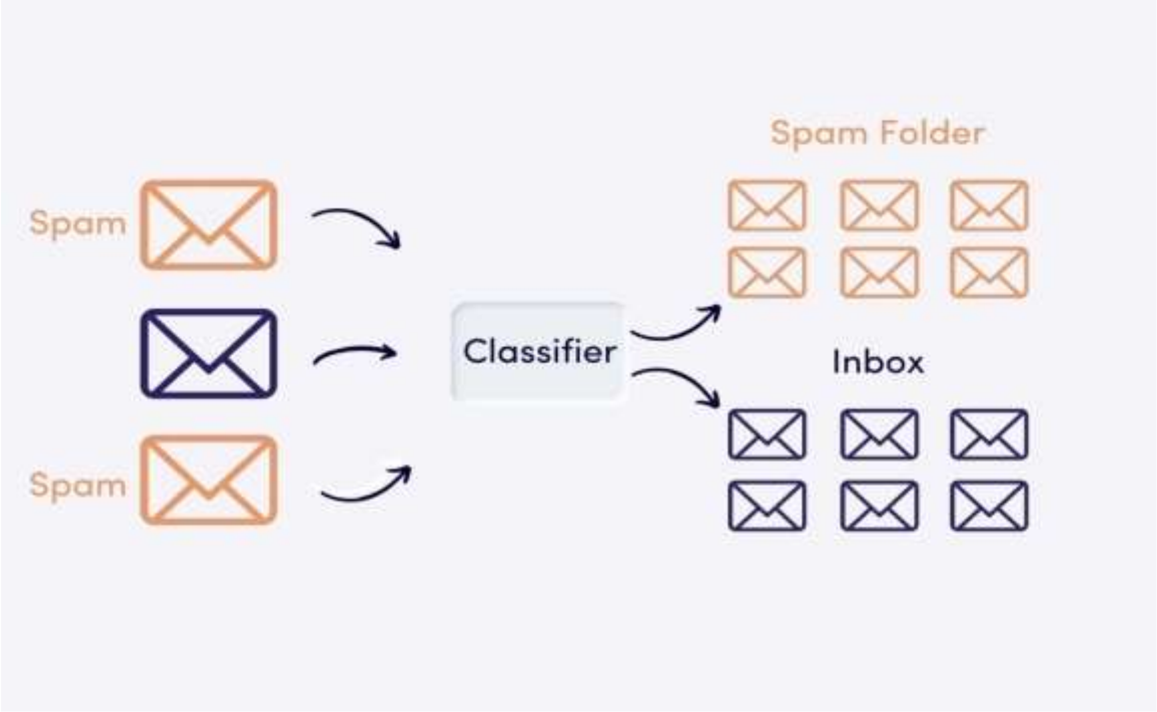


Figure 1.5: Email Classification

An NLP case study we can look at is Gmail's new classification system. This upgraded system categorizes emails into one of three groups (primary, social, or promotions) based on the email content. This is a convenient application of NLP that keeps Gmail users' inboxes under control while highlighting relevant and high-priority emails. Levity offers its own version of email classification through using NLP. This way, we can set up custom tags for our inbox and every incoming email that meets the set requirements will be sent through the correct route depending on its content [3].

1.5 How does Natural Language Processing Works?

NLP entails applying algorithms to identify and extract the natural language rules such that the unstructured language data is converted into a form that computers can understand.

When the text has been provided, the computer will utilize algorithms to extract meaning associated with every sentence and collect the essential data from them.

Sometimes, the computer may fail to understand the meaning of a sentence well, leading to obscure results.

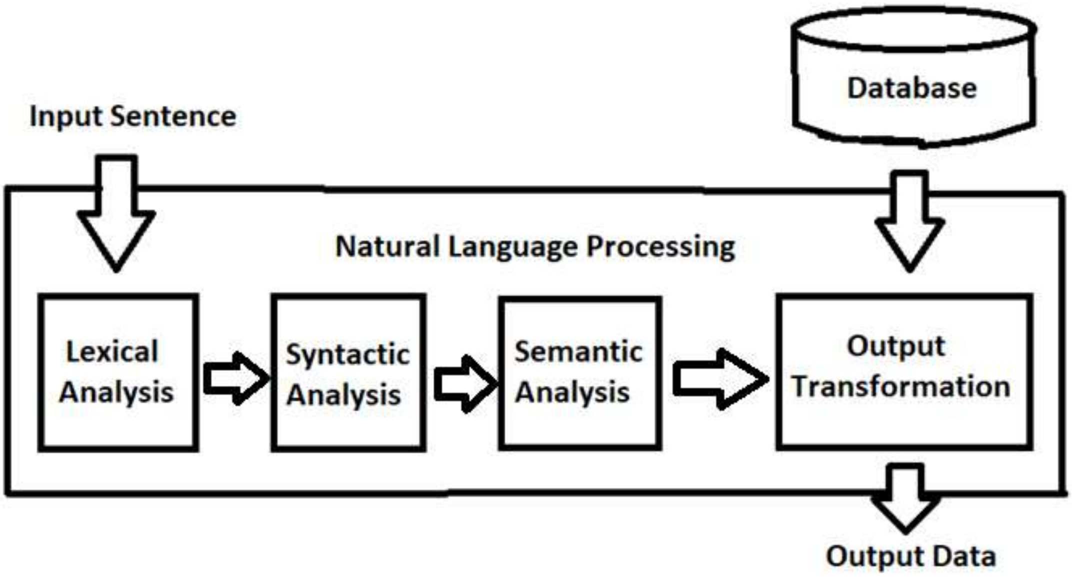


Figure 1.6: Natural Language Processing Works

1.6 Automatic Text Classification

Digitization has changed the way we process and analyze information. There is an exponential increase in online availability of information. From web pages to emails, science journals, e-books, learning content, news and social media are all full of textual data. The idea is to create, analyze and report information fast. This is when automated text classification steps up.

Text classification is a smart classification of text into categories. And using machine learning to automate these tasks just makes the whole process super-fast and efficient. Artificial Intelligence and Machine learning are arguably the most beneficial technologies to have gained momentum in recent times. They are finding applications everywhere.

Automatic text categorization is the activity of labeling text with categories from a predefined set according to their content.

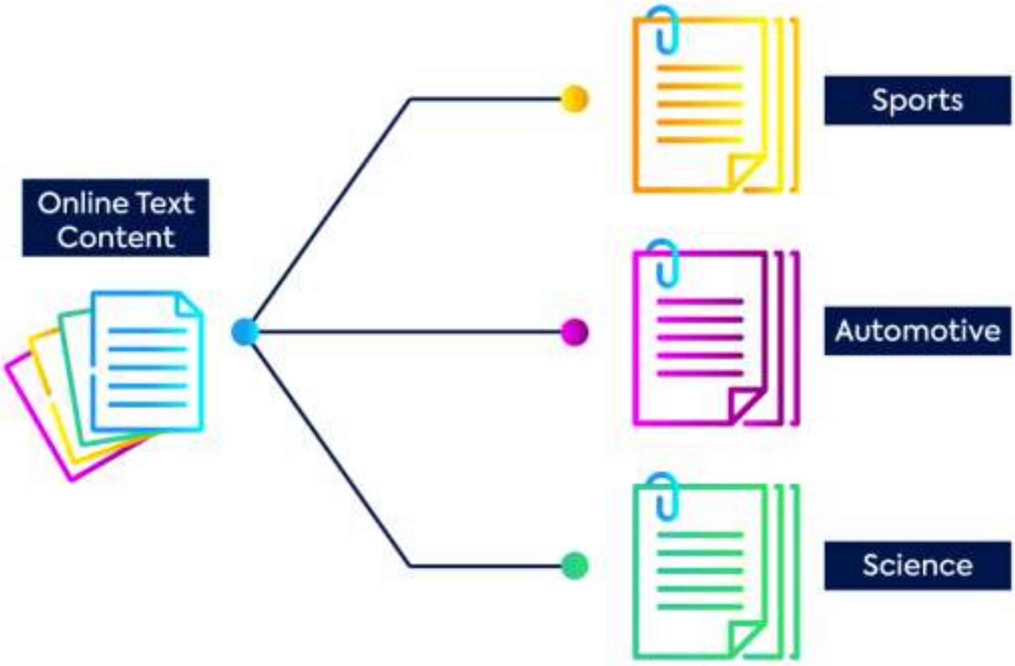


Figure 1.7: Automatic Text Classification from Raw Data

It can be applied in many contexts, from document indexing to document filtering, automated metadata generation, and any application that requires document organization or searching. Automatic text categorization helps facilitate searching documents, reports, emails, etc., that organizations manage every day, and classifying them according to their content and categories. The ability to automatically classify an article or an email into its proper category (IT, Economics, Politics, etc.) is appreciated by individual users as well as companies.

An Automatic Text Classification task can be implemented through a “rules system”, explicitly defined by a “domain expert”, or by Machine Learning systems.

Machine Learning (ML) is the ideal solution in the case where a sufficiently large set of previously classified texts is already available — a so-called “training corpus”: the corpus is supplied to the ML system, which “learns” autonomously what is the best strategy for classifying documents.

Intent, emotion and sentimental analysis of textual data are some of the most important parts of text classification. These use cases have made significant buzz among the machine intelligence enthusiasts. We have developed separate classifiers for each such category as their study is a huge topic in itself. Text classifier can operate on a variety of textual datasets. You can train the classifier with tagged data or operate on the raw unstructured text as well. Both of these categories have numerous applications of themselves.

1.6.1 Application of Text Categorization

Text Indexing

Indexing of Texts Using Controlled Vocabulary

- The documents according to the user queries, which are based on the key terms. The key terms all belong to a finite set called controlled vocabulary.
- The task of assigning keywords from a controlled vocabulary to text documents is called text indexing.

Document Sorting and Text Filtering

Document Sorting

- Sorting the given collection of documents into several “bins.”
- Examples:
- In a newspaper, the classified ads may need to be categorized into “Personal,” “Car Sale,” “Real Estate,” and so on.
- E-mail coming into an organization, which may need to be sorted into categories such as “Complaints,” “Deals,” “Job applications” and others.

Text Filtering

- Text filtering activity can be seen as document sorting with only two bins – the “relevant” and “irrelevant” documents.
- Examples:
- A sports related online magazine should filter out all non-sport stories it receives from the news feed.
- An e-mail client should filter away spam.

Web page categorization

Hierarchical Web Page Categorization

- A common use of TC is the automatic classification of Web pages under the hierarchical catalogues posted by popular Internet portals such as Yahoo.

- Constrains the number of documents belonging to a particular category to prevent the categories from becoming excessively large.
- Hyper textual nature of the documents is also another feature of the problem.

1.6.2 Enterprise Applications of Automatic Text

Document Organization: An automatic system that chooses the most suitable category for each document to classify and extract what is important from data, categorizing and making it available for more effective search and analysis.

Knowledge discovery: Traditional search and information analysis systems that apply a keyword approach will only be effective some of the time. To improve accuracy in retrieve the information we are searching for requires a technology that can understand text like we do cognitive computing applied to automatic text categorization enhances the finding ability of the data and increases the efficiency of document retrieval and accessibility.

1.6.3 The Benefits of Automatic Text Categorization

Automatic text categorization powered by cognitive technology offers companies effective information management that is independent from subjective criteria of categorization. Automatic text categorization software can identify useful information by extracting and relating relevant data present in documents. It also identifies, categorizes and makes available all sources of knowledge and reduces of search time by simplifying access to content.

1.6.4 Supervised Text Classification

Text classification models are used to categorize text into organized groups. Text is analyzed by a model and then the appropriate tags are applied based on the content. Machine learning models that can automatically apply tags for classification are known as classifiers. Classifiers can't just work automatically, they need to be trained to be able to make specific predictions for texts. Training a classifier is done by:

Training: During training, a feature extractor is used to transform each input value to a feature set. These feature sets, which capture the basic information about each input that should be used to categorize it. Pairs of feature sets and labels are fed into the machine learning algorithm to produce a model.

Prediction: During prediction, the same feature extractor is used to transform unobserved inputs to feature sets. These feature sets are then fed into the model, which produces predicted labels [4].

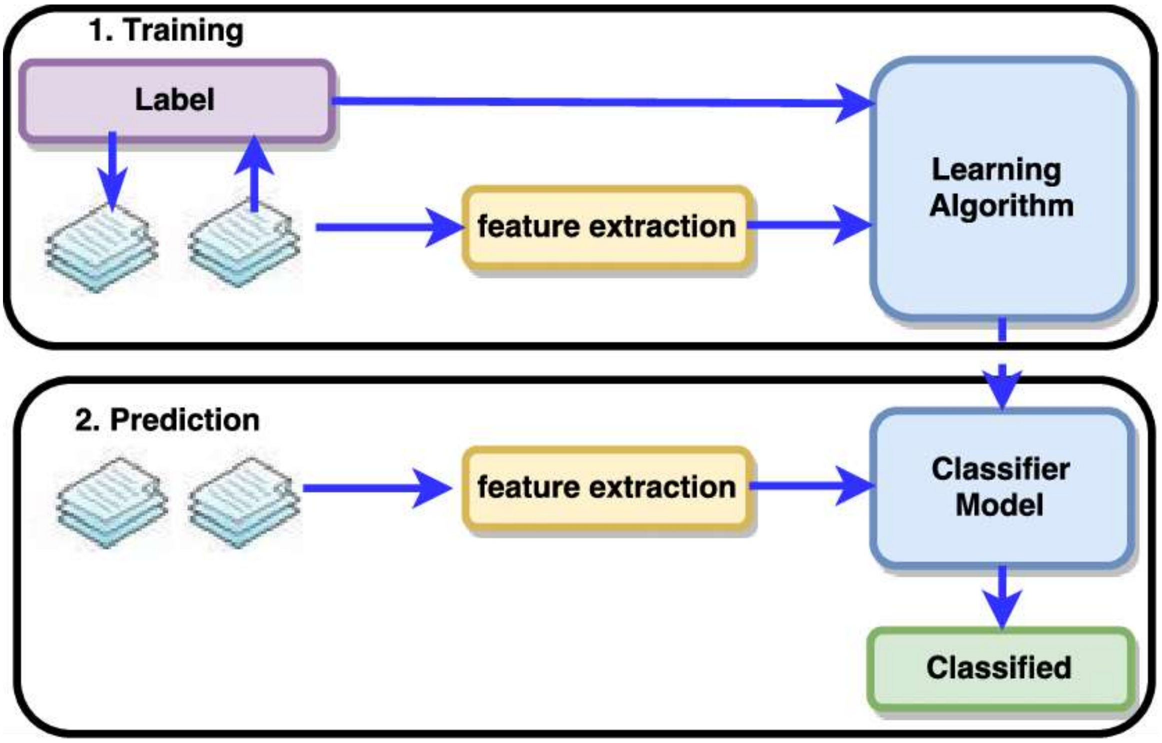


Figure 1.8: Supervised Text Classification Process

CHAPTER 2

DATA MINING

2.1Introduction

The data mining tutorial provides basic and advanced concepts of data mining. Our data mining tutorial is designed for learners and experts. Data mining is one of the most useful techniques that help entrepreneurs, researchers, and individuals to extract valuable information from huge sets of data. Data mining is also called Knowledge Discovery in Database (KDD). The knowledge discovery process includes Data cleaning, Data integration, Data selection, Data transformation, Data mining, Pattern evaluation, and Knowledge presentation. Our Data mining tutorial includes all topics of Data mining such as applications, Data mining vs Machine learning, Data mining tools, Social Media Data mining, Data mining techniques, Clustering in data mining, Challenges in Data mining, etc.

2.2 What is Data Mining?

In simple words, data mining is defined as a process used to extract usable data from a larger set of any raw data. It implies analysis data patterns in large batches of data using one or more software. Data mining has applications in multiple fields, like science and research. As an application of data mining, businesses can learn more about their customers and develop more effective strategies related to various business functions and in turn leverage resources in a more optimal and insightful manner. This helps businesses be closer to their objective and make better decisions. Data mining involves effective data collection and warehousing as well as computer processing. For segmenting the data and evaluating the probability of future events, data mining uses sophisticated mathematical algorithms. Data mining is also known as Knowledge Discovery in Data (KDD) [5].

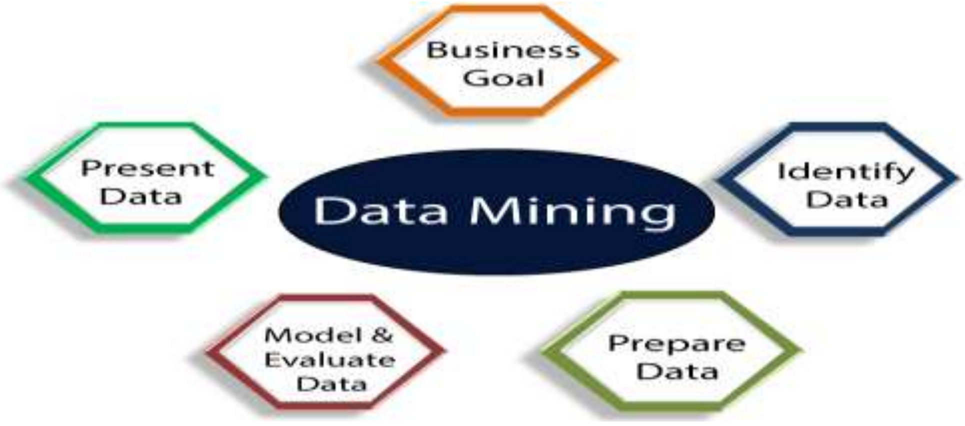


Figure 2.1: Data Mining

2.3 Types of Data Mining

Each of the following data mining techniques serves several different business problems and provides a different insight into each of them. However, understanding the type of business problem you need to solve will also help in knowing which technique will be best to use, which will yield the best results. The Data Mining types can be divided into two basic parts that are as follows:

1. Predictive Data Mining Analysis
2. Descriptive Data Mining Analysis

Predictive Data Mining: As the name signifies, Predictive Data-Mining analysis works on the data that may help to know what may happen later (or in the future) in business. Predictive Data-Mining can also be further divided into four types that are listed below:

- Classification Analysis
- Regression Analysis
- Time Serious Analysis
- Prediction Analysis

Descriptive Data Mining: The main goal of the Descriptive Data Mining tasks is to summarize or turn given data into relevant information. The Descriptive Data-Mining Tasks can also be further divided into four types that are as follows:

- Clustering Analysis
- Summarization Analysis
- Association Rules Analysis
- Sequence Discovery Analysis

2.4 Applications of Data Mining

Data is a set of discrete objective facts about an event or a process that have little use by themselves unless converted into information. We have been collecting numerous data, from simple numerical measurements and text documents to more complex information such as spatial data, multimedia channels, and hypertext documents.

Nowadays, large quantities of data are being accumulated. The amount of data collected is said to be almost doubled every year. An extracting data or seeking knowledge from this massive data, data mining techniques are used. Data mining is used in almost all places where a large amount of data is stored and processed. For example, banks typically use ‘data mining’ to find out their prospective customers who could be interested in credit cards, personal loans, or insurance as well. Since banks have the transaction details and detailed profiles of their customers, they analyze all this data and try to find out patterns that help them predict that certain customers could be interested in personal loans, etc.

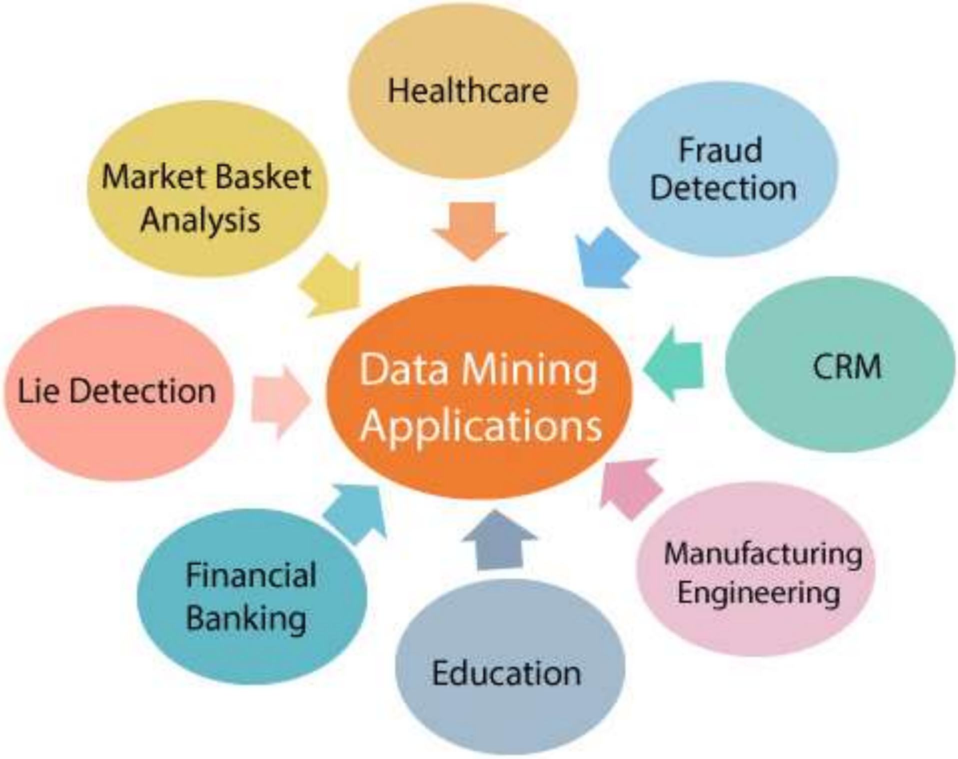


Figure 2.2: Applications of Data Mining

Technically, data mining is the computational process of analyzing data from different perspectives, dimensions, angles and categorizing/summarizing it into meaningful information. Data Mining can be applied to any type of data e.g. Data Warehouses, Transactional Databases, Relational Databases, Multimedia Databases, Spatial Databases, Time-series Databases, World Wide Web. Data mining provides competitive advantages in the knowledge economy. It does this by providing the maximum knowledge needed to rapidly make valuable business decisions despite the enormous amounts of available data [6].

2.5 Why is Data Mining Important?

Data mining helps spark ideas, thoughts, and opinions you haven’t thought of before, especially because a lot of teams are still not inclusive or diverse. Data mining gives us an outside perspective on the world and helps we make informed decisions for our business. What a graph, a marketing analytics and social media reporting company says, “Due to the massive amounts of user-generated data that is being collected and analyzed through this process, social media data mining has found wide usage and is increasingly being recognized as an invaluable asset in many fields. Although it has primarily been used for business purposes, this process is nowadays often employed by researchers and by government agencies as well." Social media marketers find data invaluable as they use it to improve the customer experience from the start of their customer journey. We can use existing data from social media to create content and help ease the issues users face daily. It's always important to do proper in-depth research before creating or refining our social media strategy. Data mining can assist with creating targeted content, products, and advertising to communicate to your social media audience. Data-driven content helps marketers to target current customers and potential prospects to improve our company's overall ROI [7].

2.6 How Does Data Mining Work?

Data mining is the process of understanding data through cleaning raw data, finding patterns, creating models, and testing those models. It includes statistics, machine learning, and database systems. Data mining often includes multiple data projects, so it's easy to confuse it with analytics, data governance, and other data processes. This guide will define data mining, share its benefits and challenges, and review how data mining works. Data mining has a long history. It emerged with computing in the 1960s through the 1980s. Historically, data mining was an intensive manual coding process — and it still involves coding ability and knowledgeable specialists to clean, process, and interpret data mining results today. Data specialists need statistical knowledge and some programming language knowledge to complete data mining techniques accurately. For instance, here are some examples of how companies have used R to answer their data questions. However, some of the manual processes are now able to be automated with repeatable flows, machine learning (ML), and artificial intelligence (AI) systems.

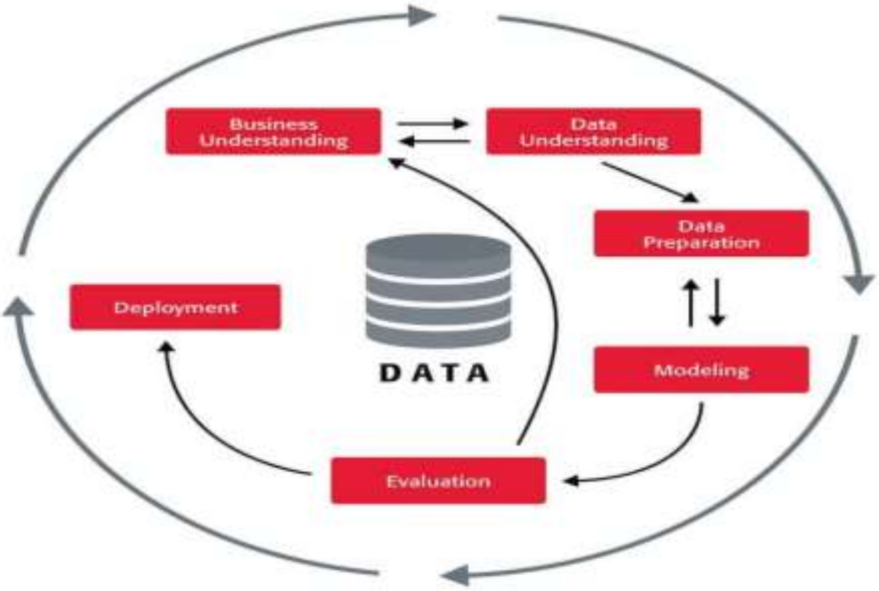


Figure 2.3: Data Mining working process

For a successful data mining process that delivers timely, reliable results, you should follow a structured, repeatable approach. Ideally, that process will include the following six steps:

Business understanding: Develop a thorough understanding of the project parameters, including the current business situation, the primary business objective of the project, and the criteria for success.

Data understanding: Determine the data that will be needed to solve the problem and gather it from all available sources.

Data preparation: Get the data ready for analysis. This includes ensuring that the data is in the appropriate format to answer the business question, and fixing any data quality problems such as missing or duplicate data.

Modeling: Use algorithms to identify patterns within the data and apply those patterns to a predictive model.

Evaluation: Determine whether and how well the results delivered by a given model will help achieve the business goal. There is often an iterative phase in which the algorithm is fine-tuned in order to achieve the best result.

Deployment: Run the analysis and make the results of the project available to decision makers.

CHAPTER 3

MACHINE LEARNING

3.1 What is Machine Learning?

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task. Machine learning is closely related to computational statistics, which focuses on making predictions using computers.

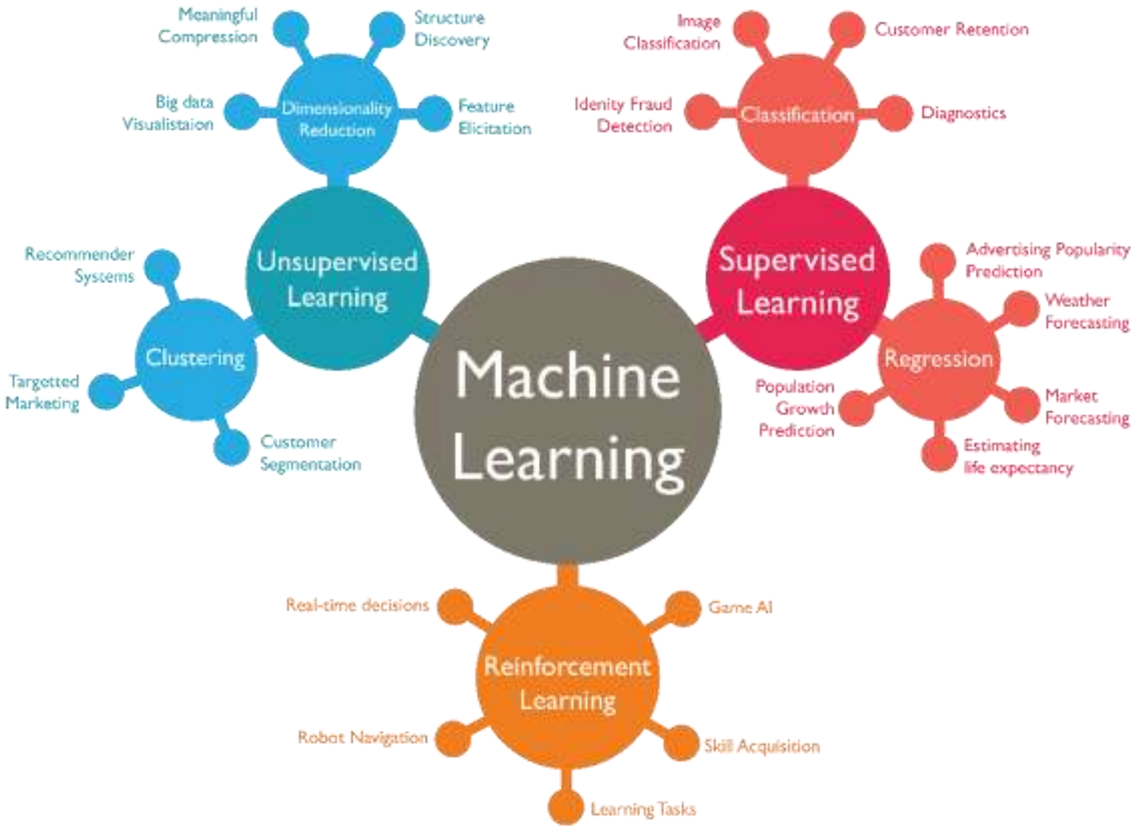


Figure 3.1: Machine Learning

3.2 Types of Machine Learning Algorithms

Based on the style and method involved, Machine Learning Algorithms are divided into four major types: Supervised Learning, Unsupervised Learning, Semi-Supervised Learning, and Reinforcement Learning.

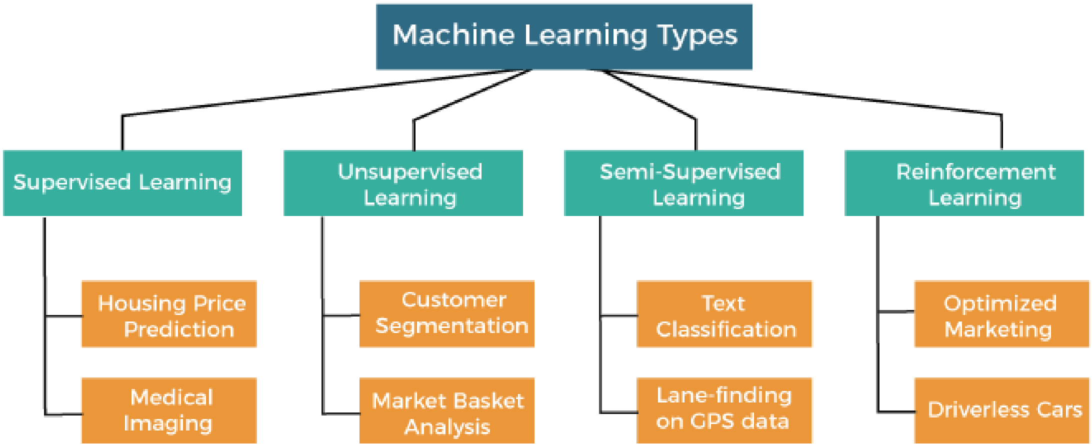


Figure 3.2: Types of Machine Learning

1.Supervised Learning: Supervised Learning is a method that involves learning using labeled past data and the algorithm shall predict the label for unseen or future data. A supervised machine learning algorithm is actually told what to look for, and so it does until it finds the underlying patterns that yield the expected output to a satisfactory degree of accuracy. In other words, using these prior known outputs, the machine learning algorithm learns from the past data and then generates an equation for the label or the value. This stage is called the training stage.

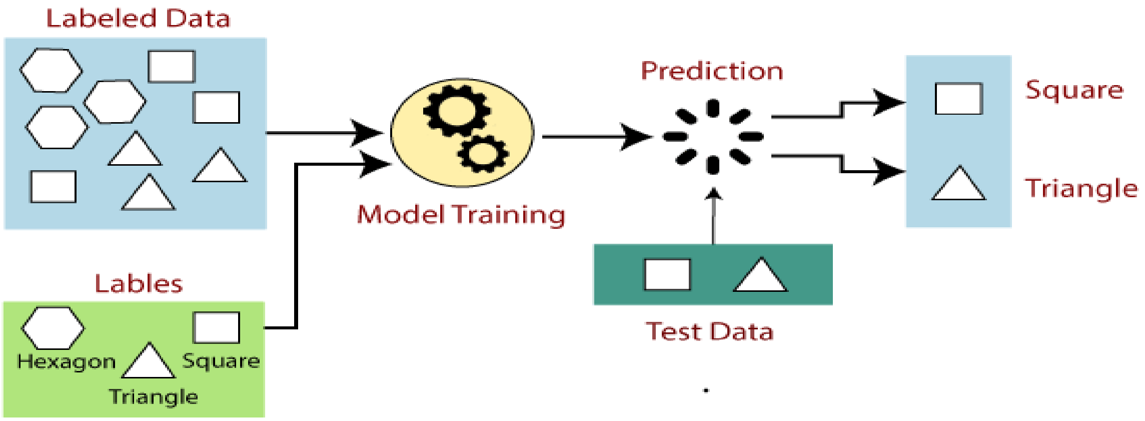


Figure 3.3: Supervised Machine Learning

The learning algorithm tries to modify and improve the above function by comparing its output with the intended, correct outputs and calculate discrepancies and errors, a task which is known as testing. During the next phase which is the implementation phase, it will take in new inputs and will generate the values or determine the label based on the generated equation.

We have seen how supervised learning algorithms can help us predict a value and an event, various forms of supervised learning algorithms are designed to solve those business problems.

i.Linear Regression: Linear Regression is a Machine Learning algorithm that maps numeric inputs to numeric outputs, by fitting a line into the data points. Simply put, Linear Regression is a way of modeling the relationship between one or more independent variables in a way that they come together to form a driving force for the dependent numerical variable. It is typically identified by the linear equation: Y=mx+C

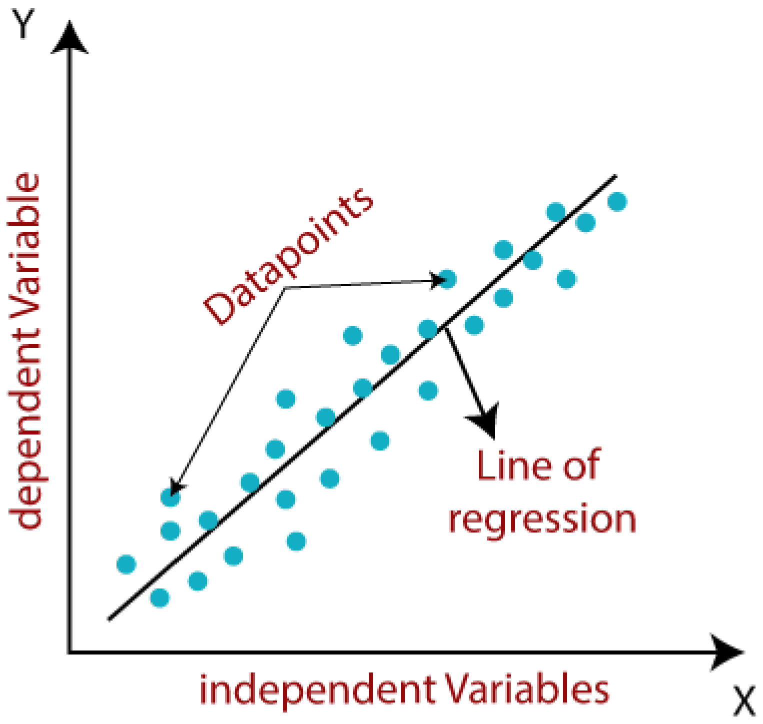


Figure 3.4: Linear Regression

ii.Logistic Regression: While Linear Regression typically works for a numerical variable, the Logistic Regression algorithm builds a relationship between variables and a class. It is typically used to predict an event class, where we have a predefined and known category of events. The dependent variable is indeed a categorical variable but the inner working of the Logistic regression algorithm actually transforms the variable by making use of a logic function, which calculates the log odds ratio for the events and hence building a linear equation for the same.

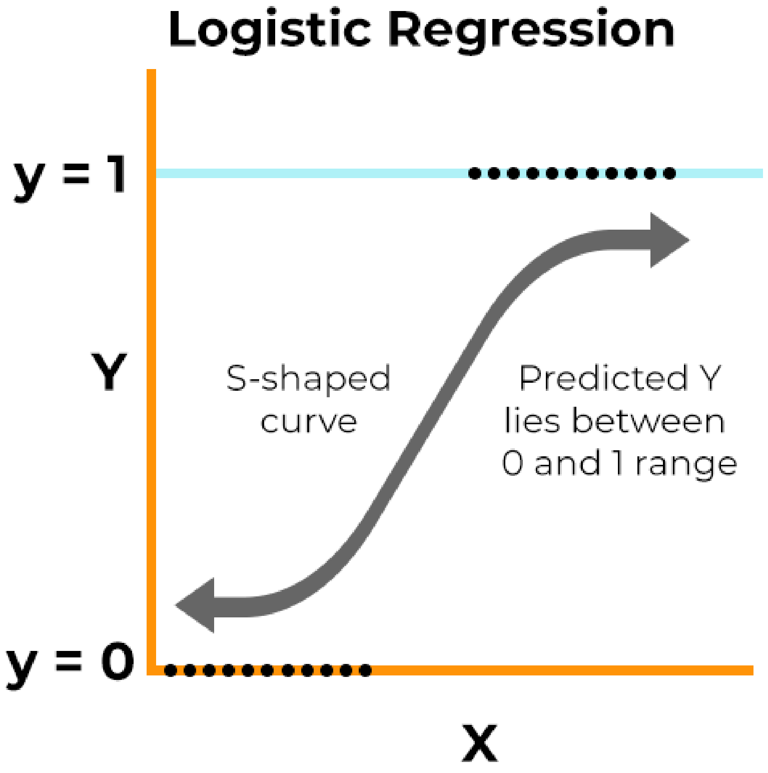


Figure 3.5: Logistic Regression

iii.Decision Trees: It is a non-parametric supervised learning technique that can be used for both Classification and Regression problems, by identifying suitable methods to split data based on various conditions into a tree-like structure. The end goal is to predict an event or a value by leveraging the obtained conditions. The tree-like structure is actually a graph where the nodes represent an underlying question about an attribute, the edges which typically contain the answers and the leaves represent the output which as we said earlier, can be a value or a class. Thus, enabling us to predict values and events. The algorithm usually follows a top-down approach, by choosing a variable at each step which can split the next set of data items and usually represented by a metric such as GINI impurity, Information Gain, Variance Reduction, etc. to measure the best approach for splitting.

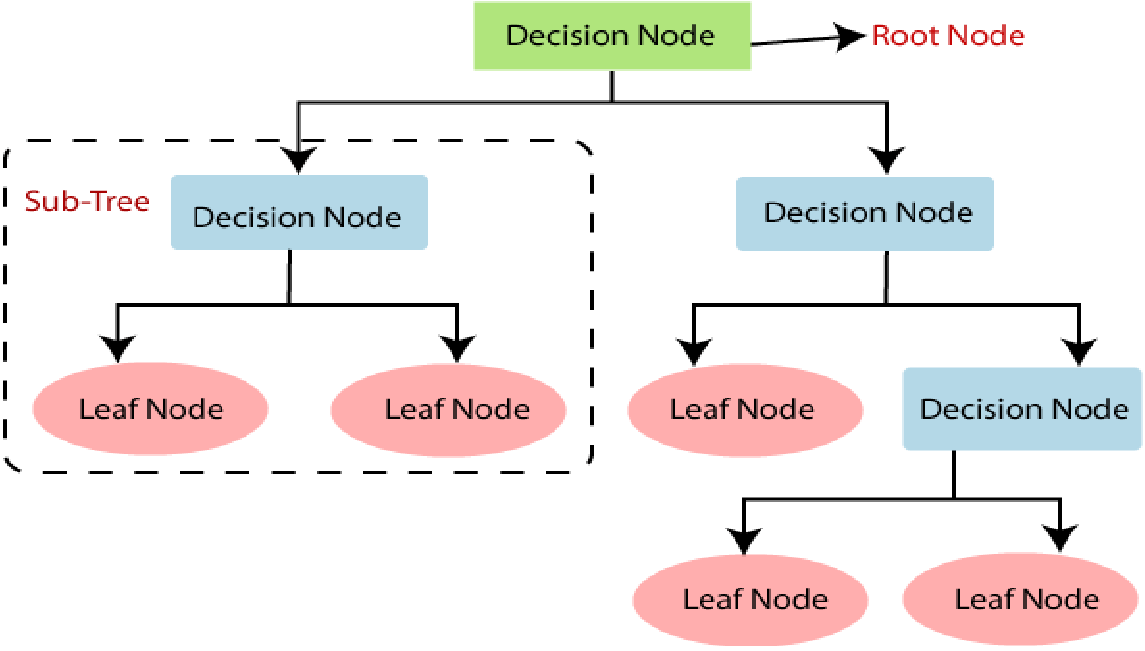


Figure 3.6: Decision Tree

iv.Support Vector Machines: This supervised machine learning algorithm is also designed for both classification and regression problems but predominantly used for Classification. It uses a technique which is known as Kernel Trick to transform the data and based on the transformation, it then finds an optimal splitting boundary between the possible outputs. The boundary can be as simple as a linear margin (Linear SVM) for binary classes, to a more complicated splitting which involves multiple classes. The algorithm represents the classes in a hyperplane in multi-dimensional space and finds the perfect divider for the classes known as a maximum marginal hyperplane.

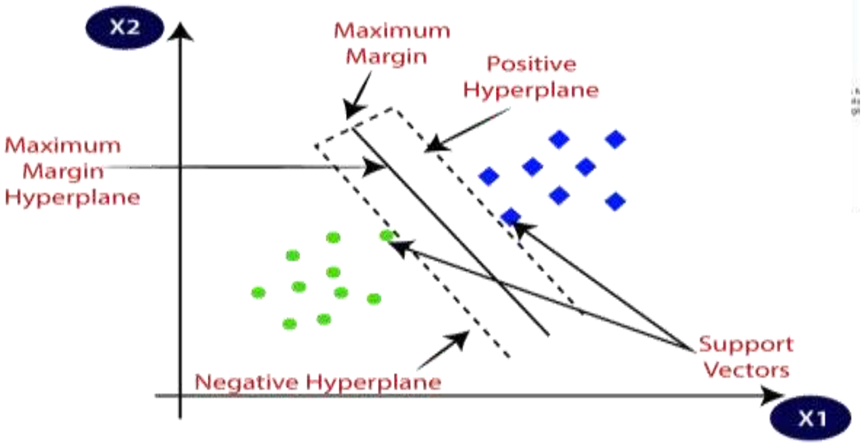


Figure 3.7: Support Vector Machine

2.Unsupervised Learning: Unsupervised learning is a type of Machine Learning algorithm that involves training a machine typically using unlabeled data, and that forms the major point of difference with supervised machine learning algorithms which typically use labeled data. In this form of machine learning, we allow the algorithm to self-discover the underlying patterns, similarities, equations, and associations in the data without adding any bias from the users' end. Although the end result of these is totally unpredictable and cannot be controlled, Unsupervised Learning finds its place is advanced exploratory data analysis and especially, Cluster Analysis. Unsupervised learning can find the hidden and unknown patterns in the data, and thus helps to find the features that are highly beneficial for auto-categorization of the data. Added to that, it is absolutely easy to get unlabelled data through various sources. For example, in a corpus of text, an unsupervised learning algorithm can get similar patterns and will be able to categorize the texts into various unknown groups – hence helping the user with discovering Topics involved in a text – like, what does a certain review about a product is talking about etc.

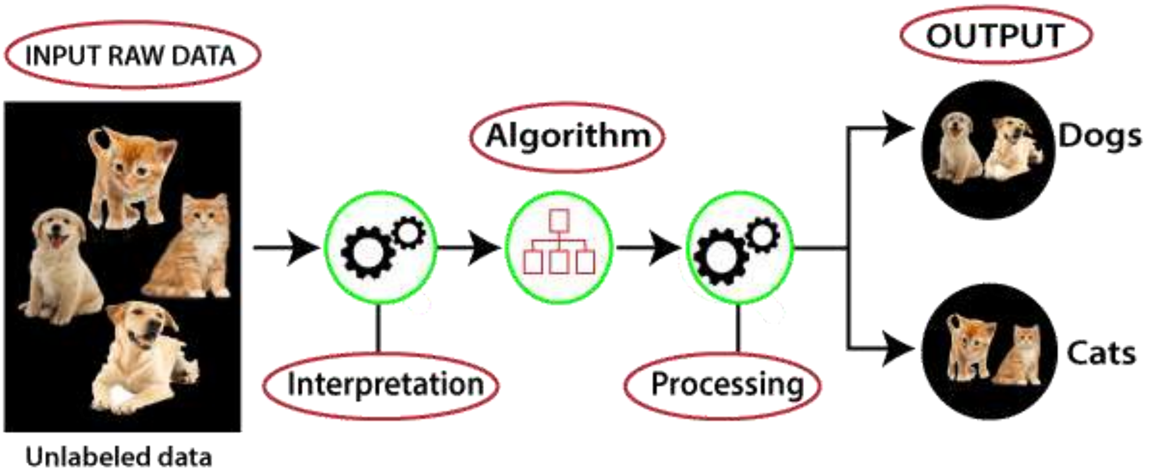


Figure 3.8: Unsupervised Machine Learning

Unsupervised learning typically involves similarity and association detections. One distinct observation about this approach is its black-box nature of the operation – where most of the input and output steps cannot be controlled by the user. In this section, let us see the most commonly used algorithms that can solve a plethora of business problems.

i.The k-means Clustering: The k-means clustering is a process that helps to partition the data points or observations into k unknown clusters in such a manner that each observation distinctly belongs to a cluster. This cluster associativity is determined by the proximity of that data point with the nearest mean, otherwise known as cluster centroid. Due to the involvement of proximity measure in the data, various distance algorithms are used in the process to measure the closeness of data to the cluster center.

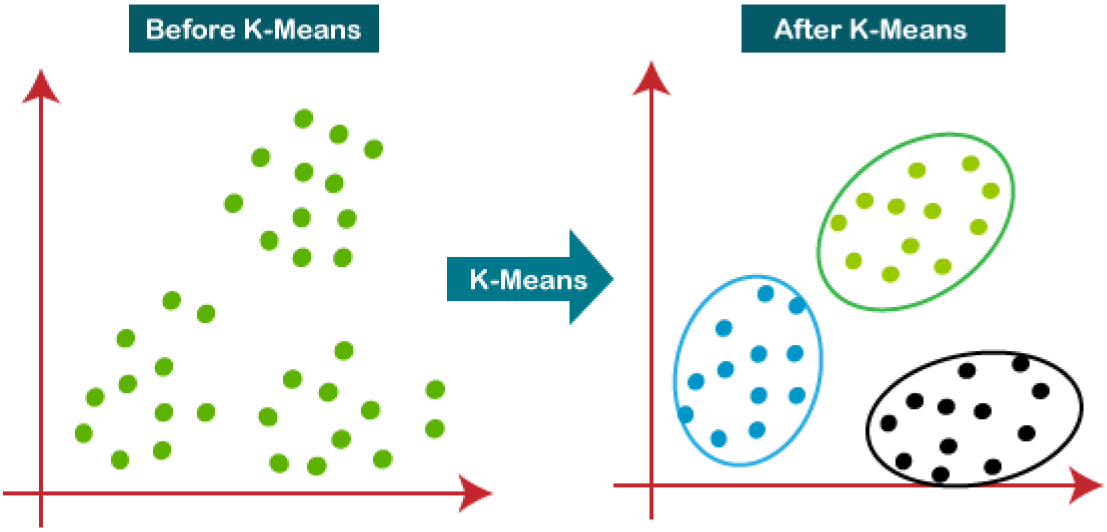


Figure 3.9: K-Means Clustering

The only and major drawback of k-means is the fact that the algorithm cannot start without the user specifying the required number of clusters apriori. Added to that, there is no mathematical or scientific method to determine the optimal number of clusters. The implementation typically occurs based on the trial and error method, where a set of "K" values are considered initially and the best one is chosen according to the heuristic knowledge.

3.Semi-Supervised Learning: Semi-Supervised learning is a type of Machine Learning algorithm that lies between Supervised and Unsupervised machine learning. It represents the intermediate ground between Supervised (With Labelled training data) and Unsupervised learning (with no labelled training data) algorithms and uses the combination of labelled and unlabeled datasets during the training period. Although Semi-supervised learning is the middle ground between supervised and unsupervised learning and operates on the data that consists of a few labels, it mostly consists of unlabeled data. As labels are costly, but for corporate purposes, they may have few labels. It is completely different from supervised and unsupervised learning as they are based on the presence & absence of labels.

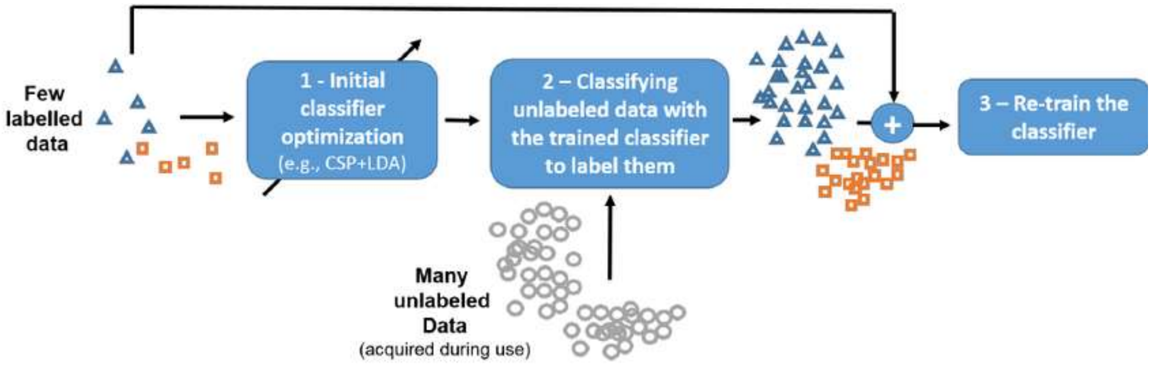

Few labelled data
1 - Initial classifier optimization (e.g., CSP+LDA)
2 – Classifying unlabeled data with the trained classifier to label them
3 – Re-train the classifier
Many unlabeled Data (acquired during use)

Figure 3.10: Semi-Supervised learning

To overcome the drawbacks of supervised learning and unsupervised learning algorithms, the concept of Semi-supervised learning is introduced. The main aim of semi-supervised learning is to effectively use all the available data, rather than only labelled data like in supervised learning. Initially, similar data is clustered along with an unsupervised learning algorithm, and further, it helps to label the unlabeled data into labelled data. It is because labelled data is a comparatively more expensive acquisition than unlabeled data.

We can imagine these algorithms with an example. Supervised learning is where a student is under the supervision of an instructor at home and college. Further, if that student is self-analysing the same concept without any help from the instructor, it comes under unsupervised learning. Under semi-supervised learning, the student has to revise himself after analyzing the same concept under the guidance of an instructor at college.

4.Reinforcement Learning: Reinforcement Learning is a subset of machine learning that enables a component of the system, called agent, to learn in a simulated, interactive virtual environment by trial and error, reinforcing the results using a reward-punishment system that is generated using feedback from its own actions and experiences.

This method sounds really close and similar to Supervised Learning but there is one major difference between them that sets them apart. We have seen how in a supervised machine learning algorithm, the program knows the answers and accordingly it designs patterns and creates a model. In Reinforcement Learning, however, there are no predefined labels. It operates inside of a virtual environment, which is supplemented by a set of rewards for correct answers and a set of punishments for incorrect answers. The goal of the algorithm is ultimately to maximize the rewards for the software generated agent.

In this technique, the model keeps on increasing its performance using Reward Feedback to learn the behavior or pattern. These algorithms are specific to a particular problem e.g. Google Self Driving car, AlphaGo where a bot competes with humans and even itself to get better and better performers in Go Game.

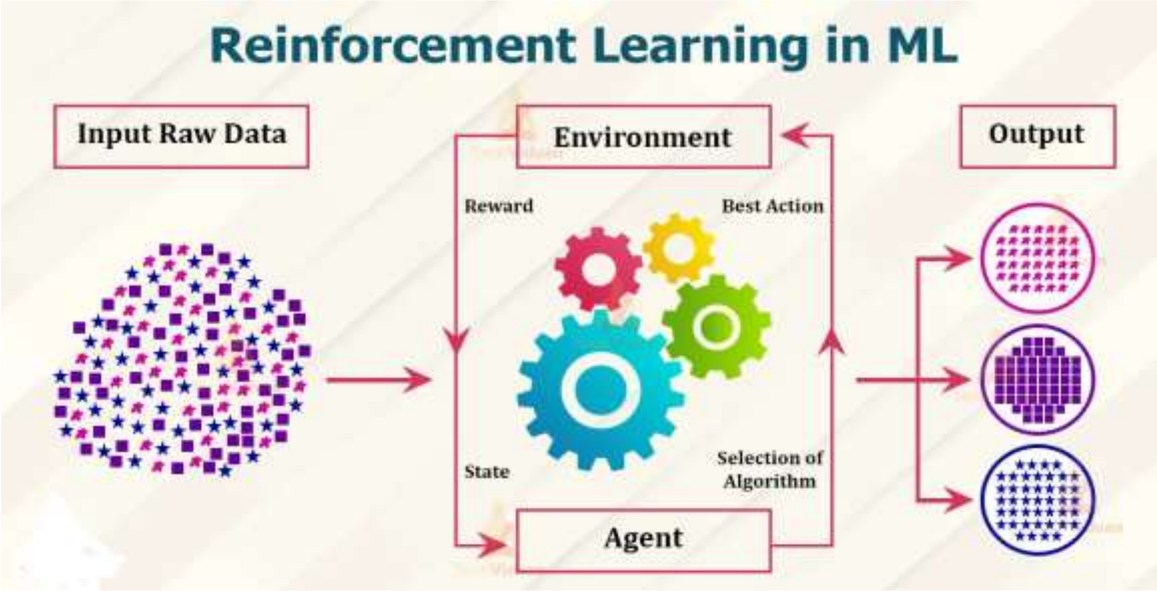


Figure 3.11: Reinforcement Learning

Each time we feed in data, they learn and add the data to their knowledge which is training data. So, the more it learns the better it gets trained and hence experienced.

i. Agents observe input.
ii. An agent performs an action by making some decisions.
iii. After its performance, an agent receives a reward and accordingly reinforces and
iv. the model stores in state-action pair of information.
v. Temporal Difference (TD)
vi. Q-Learning
vi. Deep Adversarial Networks [8]

3.3How Does Data Machine Learning works

Machine Learning is, undoubtedly, one of the most exciting subsets of Artificial Intelligence. It completes the task of learning from data with specific inputs to the machine. It’s important to understand what makes Machine Learning work and, thus, how it can be used in the future. The Machine Learning process starts with inputting training data into the selected algorithm. Training data being known or unknown data to develop the final Machine Learning algorithm. The type of training data input does impact the algorithm, and that concept will be covered further momentarily. New input data is fed into the machine learning algorithm to test whether the algorithm works correctly. The prediction and results are then checked against each other. If the prediction and results don’t match, the algorithm is re-trained multiple times until the data scientist gets the desired outcome. This enables the machine learning algorithm to continually learn on its own and produce the optimal answer, gradually increasing in accuracy over time.

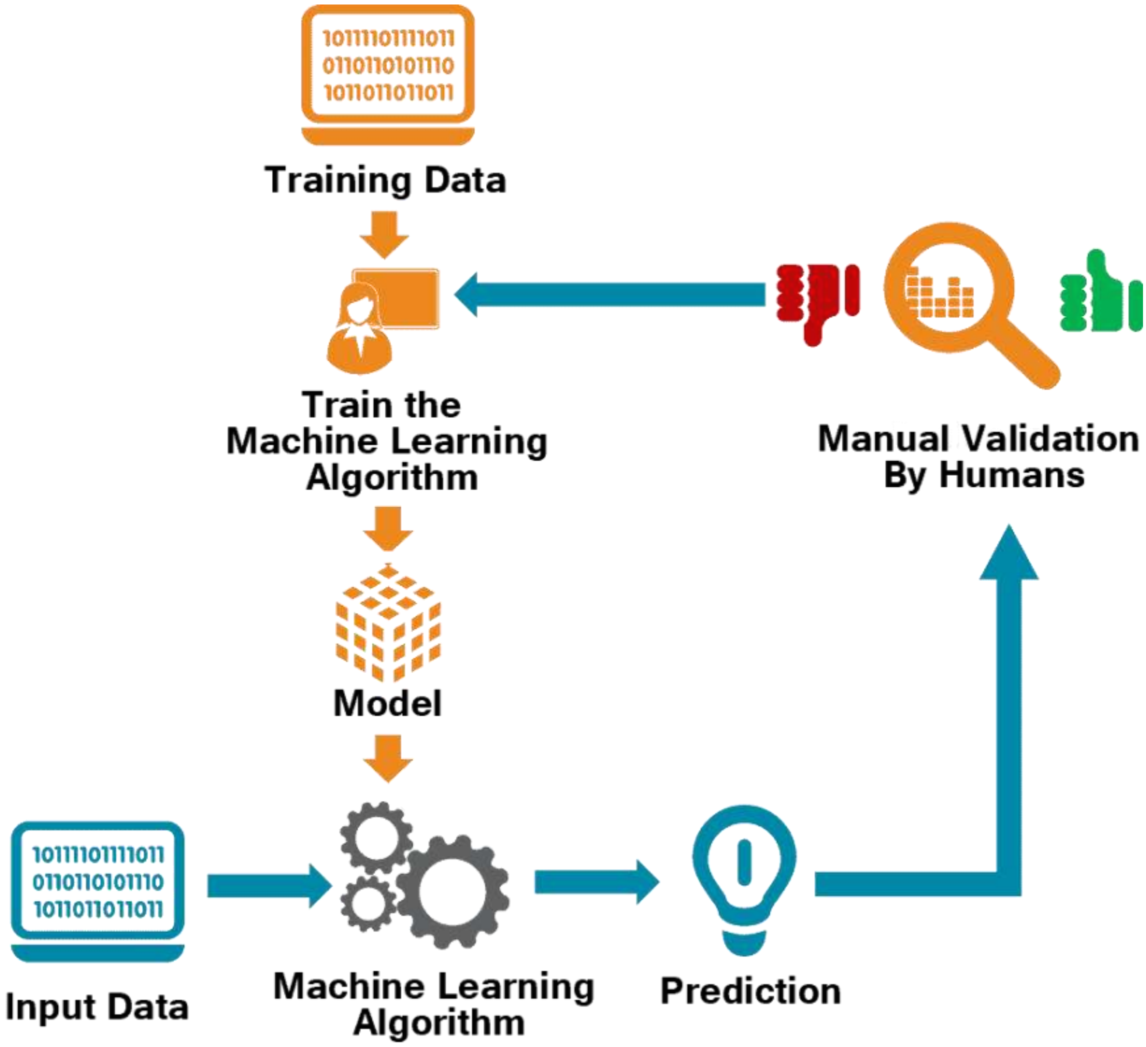


Figure 3.12: Machine Learning Work

3.4Applications of Machine Learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

Few applications are described below.

Image Recognition: Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, etc. The popular use case of image recognition and face detection is, Facebook provides us a feature of auto friend tagging suggestion. Whenever we upload a photo with our Facebook friends, then we automatically get a tagging suggestion with name, and the technology behind this is machine learning's face detection and recognition algorithm.

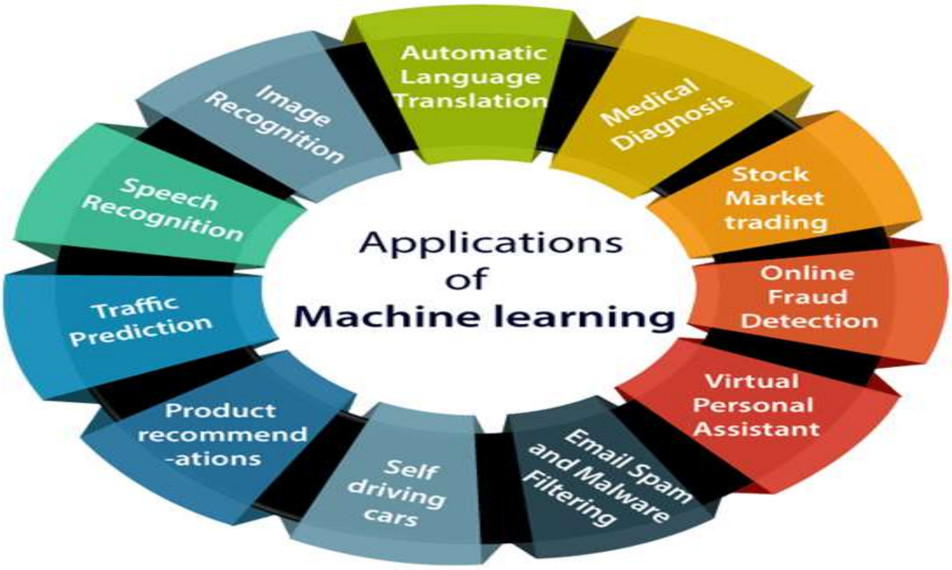

Applications of Machine learning
Automatic Language Translation
Medical Diagnosis
Stock Market trading
Online Fraud Detection
Virtual Personal Assistant
Email Spam and Malware Filtering
Self driving cars
Product recommend -ations
Traffic Prediction
Speech Recognition
Image Recognition

Figure 3.13: Applications of Machine Learning

Speech Recognition: While using Google, we get an option of "Search by voice," it comes under speech recognition, and it's a popular application of machine learning. Speech recognition is a process of converting voice instructions into text, and it is also known as "Speech to text", or "Computer speech recognition.

Traffic Prediction: If we want to visit a new place, we take help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions. It predicts the traffic conditions such as whether traffic is cleared, slow-moving, or heavily congested with the help of two ways: (1) Real Time location of the vehicle form Google Map app and sensors. (2) Average time has taken on past days at the same time.
Everyone who is using Google Map is helping this app to make it better. It takes information from the user and sends back to its database to improve the performance.

Product Recommendations: Machine learning is widely used by various e-commerce and entertainment companies such as Amazon, Netflix, etc., for product recommendation to the user. Whenever we search for some product on Amazon, then we started getting an advertisement for the same product while internet surfing on the same browser and this is because of machine learning.

Self-Driving Cars: Machine learning plays a significant role in self-driving cars. Tesla, the most popular car manufacturing company is working on self-driving car. It is using unsupervised learning method to train the car models to detect people and objects while driving.

Email Spam and Malware Filtering: Whenever we receive a new email, it is filtered automatically as important, normal, and spam. Below are some spam filters used by Gmail: Content Filter C, Header Filter, General Blacklist Filter, Rule Based Filter, Permission Filter, etc.

Virtual Personal Assistant: We have various virtual personal assistants such as Google assistant, Alexa, Cortana, Siri. As the name suggests, they help us in finding the information using our voice instruction. These assistants can help us in various ways just by our voice instructions such as Play music, call someone, Open an email, Scheduling an appointment, etc. These virtual assistants use machine learning algorithms as an important part.

Online Fraud Detection: Machine learning is making our online transaction safe and secure by detecting fraud transaction. Whenever we perform some online transaction, there may be various ways that a fraudulent transaction can take place such as fake accounts, fake ids, and steal money in the middle of a transaction. So to detect this, Feed Forward Neural network helps us by checking whether it is a genuine transaction or a fraud transaction.

Stock Market Trading: Machine learning is widely used in stock market trading. In the stock market, there is always a risk of up and downs in shares, so for this machine learning's long short term memory neural network is used for the prediction of stock market trends.

Medical Diagnosis: In medical science, machine learning is used for diseases diagnoses. With this, medical technology is growing very fast and able to build 3D models that can predict the exact position of lesions in the brain. It helps in finding brain tumors and other brain-related diseases easily.

Automatic Language Translation: Nowadays, if we visit a new place and we are not aware of the language then it is not a problem at all, as for this also machine learning helps us by converting the text into our known languages. Google's GNMT (Google Neural Machine Translation) provide this feature, which is a Neural Machine Learning that translates the text into our familiar language, and it called as automatic translation [9].

3.5 Why is Machine Learning Important?

Machine learning is a form of artificial intelligence (AI) that teaches computers to think in a similar way to humans: learning and improving upon past experiences. Almost any task that can be completed with a data-defined pattern or set of rules can be automated with machine learning.

So, why is machine learning important? It allows companies to transform processes that were previously only possible for humans to perform think responding to customer service calls, bookkeeping, and reviewing resumes for everyday businesses. Machine learning can also scale to handle larger problems and technical questions think image detection for self-driving cars, predicting natural disaster locations and timelines, and understanding the potential interaction of drugs with medical conditions before clinical trials. That's why machine learning is important.

3.6 Example of Some Classical Machine Learning Algorithm

Linear Regression: If you want to start machine learning, Linear regression is the best place to start. Linear Regression is a regression model, meaning, it'll take features and predict a continuous output, stock price, salary etc. LR allocates weight parameter, theta for each of the training features. The predicted output (h(θ)) will be a linear function of features and θ coefficients.

$$h_\theta = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots$$

During the start of training, each theta is randomly initialized. But during the training, we correct the theta corresponding to each feature such that, the loss (metric of the deviation between expected and predicted output) is minimized. Gradient descend algorithm will be used to align the θ values in the right direction. In the below diagram, each red dots represent the training data and the blue line shows the derived solution.

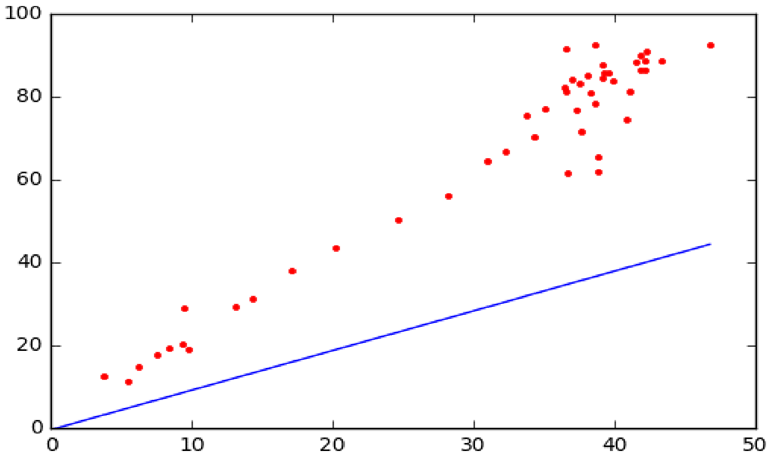


Figure 3.14: Linear Regression

Loss Function: In LR, we use mean squared error as the metric of loss. The deviation of expected and actual outputs will be squared and sum up. Derivative of this loss will be used by gradient descend algorithm.

Logistic Regression: Logistic regression is the right algorithm to start with classification algorithms. Even though, the name 'Regression' comes up, it is not a regression model, but a classification model. It uses a logistic function to frame binary output model. The output of the logistic regression will be a probability ($0 \leq x \leq 1$), and can be used to predict the binary 0 or 1 as the output (if $x<0.5$, output= 0, else output=1). Logistic Regression acts somewhat very similar to linear regression. It also calculates the linear output, followed by a stashing function over the regression output. Sigmoid function is the frequently used logistic function. You can see below clearly, that the z value is same as that of the linear regression output in equation (1).

$$z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots.$$

$$h(\theta) = g(z)$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

The h(θ) value here corresponds to P(y=1|x), i.e, probability of output to be binary 1, given input x. P(y=0|x) will be equal to 1-h(θ).when value of z is 0, g(z) will be 0.5. Whenever z is positive, h (θ) will be greater than 0.5 and output will be binary 1. Likewise, whenever z is negative, value of y will be 0. As we use a linear equation to find the classifier, the output model also will be a linear one that means it splits the input dimension into two spaces with all points in one space corresponds to same label.

The figure below shows the distribution of a sigmoid function:

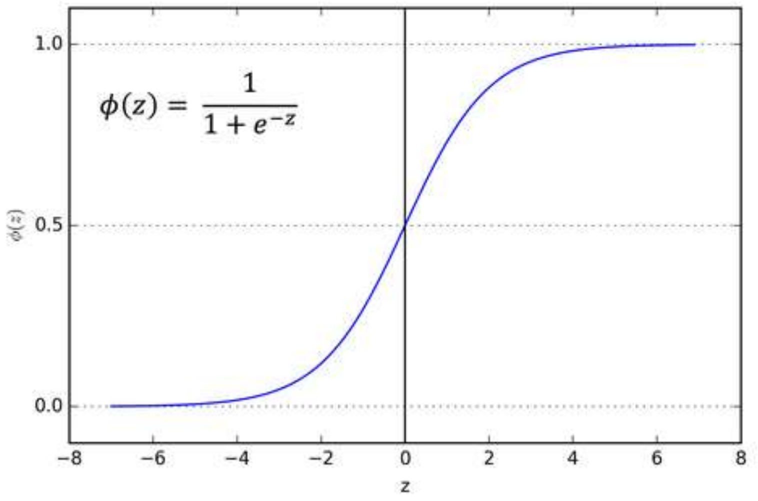


Figure 3.15: Logistic Regression

Loss function: We can’t use mean squared error as loss function(like linear regression), because we use a non-linear sigmoid function at the end. MSE function may introduce local minimums and will affect the gradient descend algorithm.

So we use cross entropy as our loss function here. Two equations will be used, corresponding to y=1 and y=0. The basic logic here is that, whenever my prediction is badly wrong, (eg : y’ =1 & y = 0), cost will be -log(0) which is infinity.

$$J(\theta) = \frac{1}{m} \sum cost(y', y)$$

$$cost(y', y) = -log(1 - y') \;\; if \;\; y = 0$$

$$cost(y', y) = -log(y') \;\; if \;\; y = 1$$

In the equation given, m stands for training data size, y’ stands for predicted output and y stands for actual output [10].

CHAPTER 4

FEATURE EXTRACTION

4.1 What is Feature Extraction?

Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing. A characteristic of these large data sets is a large number of variables that require a lot of computing resources to process. Feature extraction is the name for methods that select and /or combine variables into features, effectively reducing the amount of data that must be processed, while still accurately and completely describing the original data set. If the number of features becomes similar (or even bigger!) than the number of observations stored in a dataset then this can most likely lead to a Machine Learning model suffering from overfitting. In order to avoid this type of problem, it is necessary to apply either regularization or dimensionality reduction techniques (Feature Extraction). In Machine Learning, the dimensionality of a dataset is equal to the number of variables used to represent it[11].

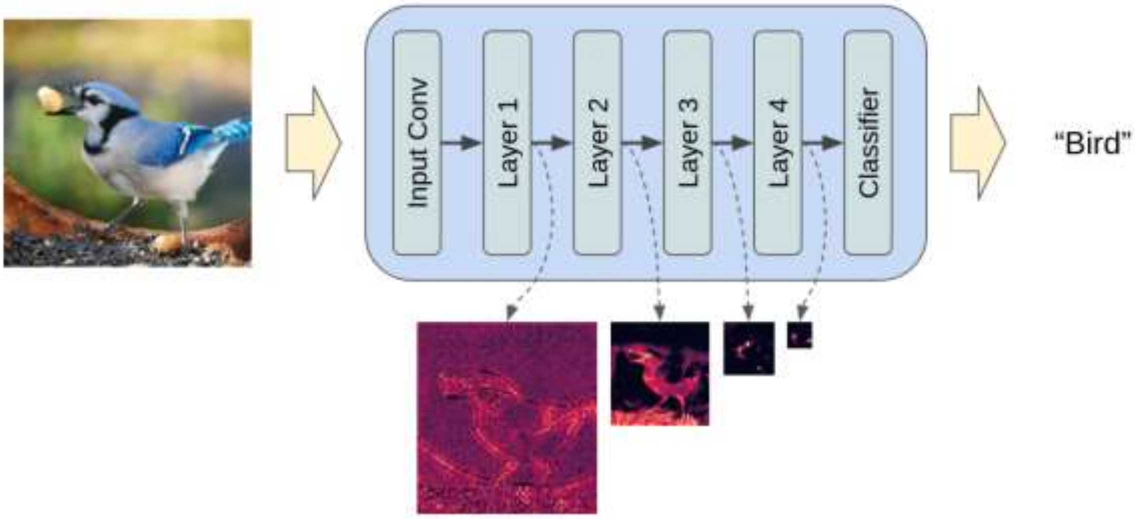


Figure 4.1: Feature Extraction

4.2 Why is this Useful?

The process of feature extraction is useful when you need to reduce the number of resources needed for processing without losing important or relevant information. Feature extraction can also reduce the amount of redundant data for a given analysis. Also, the reduction of the data and the machine's efforts in building variable combinations (features) facilitate the speed of learning and generalization steps in the machine learning process.

4.3 Bag of Words (BOW)

A bag-of-words model, or BOW for short, is a way of extracting features from text for use in modeling, such as with machine learning algorithms. The approach is very simple and flexible, and can be used in a myriad of ways for extracting features from documents. A bag-of-words is a representation of text that describes the occurrence of words within a document. It involves two things:

i)A vocabulary of known words.
ii)A measure of the presence of known words.

It is called a *"*bag*"* of words, because any information about the order or structure of words in the document is discarded. The model is only concerned with whether known words occur in the document, not where in the document. A very common feature extraction procedures for sentences and documents is the bag-of-words approach (BOW). In this approach, we look at the histogram of the words within the text, i.e. considering each word count as a feature. The intuition is that documents are similar if they have similar content. Further, that from the content alone we can learn something about the meaning of the document. The bag-of-words can be as simple or complex as you like. The complexity comes both in deciding how to design the vocabulary of known words (or tokens) and how to score the presence of known words [12].

4.3.1 Example of Bag of Words (BOW)

Let’s make the bag-of-words model concrete with a worked example.

Step 1 (Collect Data): Below is a snippet of the first few lines of text from the book “A Tale of Two Cities” by Charles Dickens, taken from Project Gutenberg.

It is a flower,
It is a Rose flower,
It is a black Rose flower,
It is a worst Rose flower,

For this small example, let’s treat each line as a separate “document” and the 4 lines as our entire corpus of documents.

Step 2 (Design the Vocabulary): Now we can make a list of all of the words in our model vocabulary. The unique words here (ignoring case and punctuation) are:

"It"
"is"
"a"
"flower"
"Rose"
"black"
"worst"

That is a vocabulary of 7 words from a corpus containing 21 words.

Step 3 (Create Document Vectors): The next step is to score the words in each document. The objective is to turn each document of free text into a vector that we can use as input or output for a machine learning model. Because we know the vocabulary has 7 words, we can use a fixed-length document representation of 7, with one position in the vector to score each word. The simplest scoring method is to mark the presence of words as a boolean value, 0 for absent, 1 for present. Using the arbitrary ordering of words listed above in our vocabulary, we can step through the first document ("It is a flower") and convert it into a binary vector. The scoring of the document would look as follows:

"It"= 1
"is" =1
"a" =1
"flower" =1
"Rose" =0
"black" =0
"worst"= 0

As a binary vector, this would look as follows:[1, 1, 1, 1, 0, 0, 0]
The other three documents would look as follows:

" It is a Rose flower" = [1, 1, 1, 0, 1, 0, 0]
" It is a black Rose flower" = [1, 1, 1, 0,0, 1, 0]
" It is a worst Rose flower" = [1, 1, 1, 0, 0, 1, 0]

All ordering of the words is nominally discarded and we have a consistent way of extracting features from any document in our corpus, ready for use in modeling. New documents that overlap with the vocabulary of known words, but may contain words outside of the vocabulary, can still be encoded, where only the occurrence of known words are scored and unknown words are ignored [13].

4.3.2 Limitations of Bag of Words

The bag-of-words model is very simple to understand and implement and offers a lot of flexibility for customization on your specific text data. It has been used with great success on prediction problems like language modeling and documentation classification. Nevertheless, it suffers from some shortcomings, such as:

Vocabulary: The vocabulary requires careful design, most specifically in order to manage the size, which impacts the sparsity of the document representations.

Sparsity: Sparse representations are harder to model both for computational reasons (space and time complexity) and also for information reasons, where the challenge is for the models to harness so little information in such a large representational space.

Meaning: Discarding word order ignores the context, and in turn meaning of words in the document (semantics). Context and meaning can offer a lot to the model, that if modeled could tell the difference between the same words differently arranged (“this is interesting” vs “is this interesting”), synonyms (“old bike” vs “used bike”), and much more [14].

CHAPTER 5

DIFFERENT TYPES OF CLASSIFIERS

5.1 What is Classifier?

A classifier in machine learning is an algorithm that automatically orders or categorizes data into one or more of a set of "classes." One of the most common examples is an email classifier that scans emails to filter them by class label: Spam or Not Spam. Machine learning algorithms are helpful to automate tasks that previously had to be done manually. They can save huge amounts of time and money and make businesses more efficient.

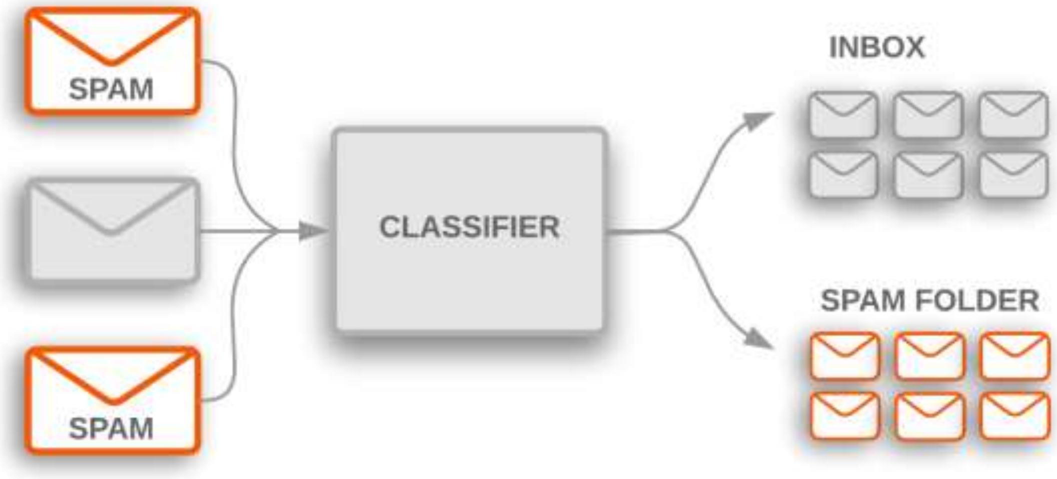


Figure 5.1: Classifier

For example, spam detection in email service providers can be identified as a classification problem. This is s binary classification since there are only 2 classes as spam and not spam. A classifier utilizes some training data to understand how given input variables relate to the class. In this case, known spam and non-spam emails have to be used as the training data. When the classifier is trained accurately, it can be used to detect an unknown email.

In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc.

5.2 Classic Machine Learning

Machine Learning (ML) is arguably the most important field of Artificial Intelligence today. It refers to the process of building algorithms that can learn from existing observations (or data sets), and leverage that learning to predict new observations, or determine the output of new input. The significance of Machine Learning lies in the fact that all other fields within AI (say, Computer Vision or Natural Language Processing) generally rely on Machine Learning to achieve their intended objectives.

Course5’s global team of Data Scientists and Machine Learning engineers empower organizations to adopt and apply advanced Machine Learning algorithms and techniques in their digital transformation journeys, and always stay ahead of their competition.

5.2.1 Logistic Regression

Logistic regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on. Multinomial logistic regression can model scenarios where there are more than two possible discrete outcomes. Logistic regression is a useful analysis method for classification problems, where you are trying to determine if a new sample fits best into a category. As aspects of cyber security are classification problems, such as attack detection, logistic regression is a useful analytic technique.

5.2.1.1 What is Logistic Regression?

Logistic regression is another powerful supervised ML algorithm used for binary classification problems (when target is categorical). The best way to think about logistic regression is that it is a linear regression but for classification problems. Logistic regression essentially uses a logistic function defined below to model a binary output variable (Tolles & Meurer, 2016). The primary difference between linear regression and logistic regression is that logistic regression's range is bounded between 0 and 1. In addition, as opposed to linear regression, logistic regression does not require a linear relationship between inputs and output variables. This is due to applying a nonlinear log transformation to the odds ratio (will be defined shortly).

$$Logistic\ function = \frac{1}{1+e^{-x}}$$

In the logistic function equation, *x* is the input variable. Let's feed in values −20 to 20 into the logistic function. As illustrated in Fig. 5.2, the inputs have been transferred to between 0 and 1.

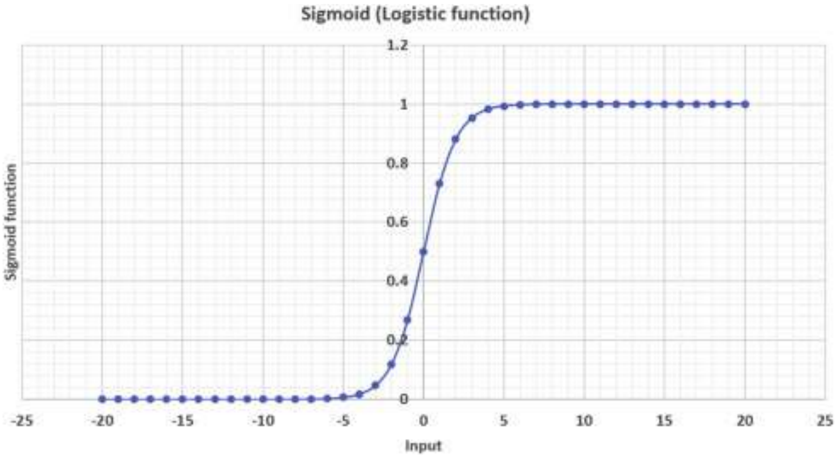


Figure 5.2: Logistic Function

5.2.1.2 What is Logistic Regression used for?

Logistic regression is used to calculate the probability of a binary event occurring, and to deal with issues of classification. For example, predicting if an incoming email is spam or not spam, or predicting if a credit card transaction is fraudulent or not fraudulent. In a medical context, logistic regression may be used to predict whether a tumor is benign or malignant. In marketing, it may be used to predict if a given user (or group of users) will buy a certain product or not. An online education company might use logistic regression to predict whether a student will complete their course on time or not.

As you can see, logistic regression is used to predict the likelihood of all kinds of "yes" or "no" outcomes. By predicting such outcomes, logistic regression helps data analysts (and the companies they work for) to make informed decisions. In the grand scheme of things, this helps to both minimize the risk of loss and to optimize spending in order to maximize profits. And that's what every company wants, right?

For example, it wouldn't make good business sense for a credit card company to issue a credit card to every single person who applies for one. They need some kind of method or model to work out, or predict, whether or not a given customer will default on their payments. The two possible outcomes, "will default" or "will not default", comprise binary data—making this an ideal use-case for logistic regression. Based on what category the customer falls into, the credit card company can quickly assess who might be a good candidate for a credit card and who might not be. Similarly, a cosmetics company might want to determine whether a certain customer is likely to respond positively to a promotional 2-for-1 offer on their skincare range. In which case, they may use logistic regression to devise a model which predicts whether the customer will be a "responder" or a "non-responder." Based on these insights, they'll then have a better idea of where to focus their marketing efforts.

5.2.1.3 What are the different types of Logistic Regression

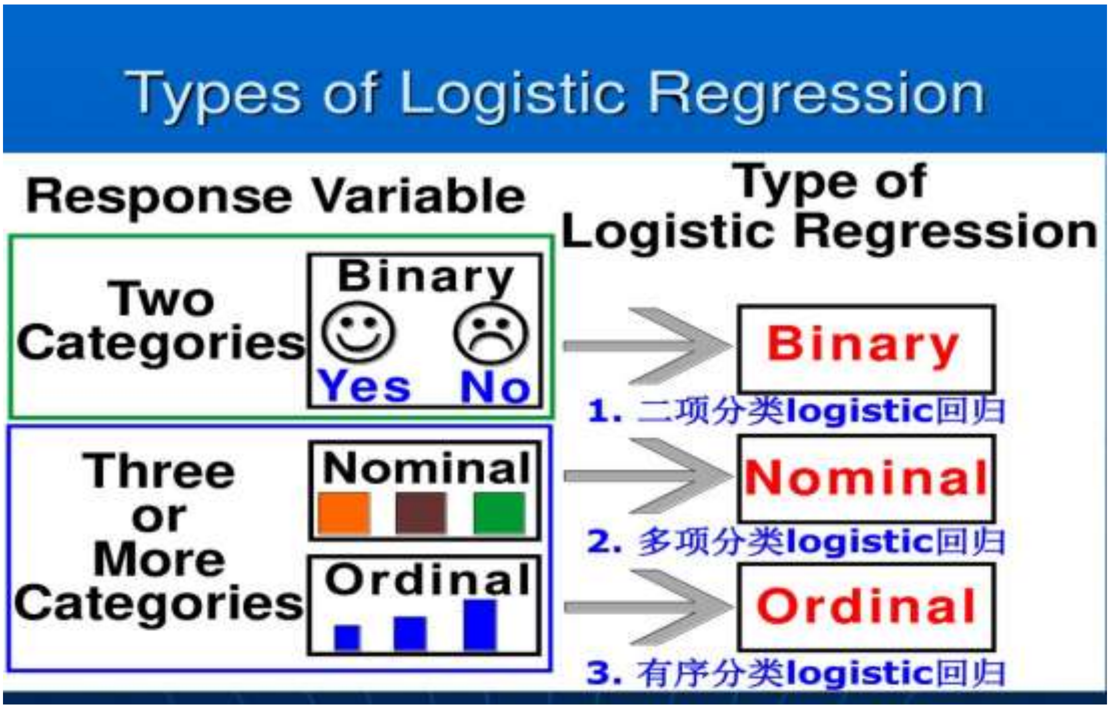
There are three main types of logistic regression: binary, multinomial and ordinal. They differ in execution and theory. Binary regression deals with two possible values, essentially: yes or no. Multinomial logistic regression deals with three or more values. And ordinal logistic regression deals with three or more classes in a predetermined order.

1.Binary logistic regression: Binary logistic regression was mentioned earlier in the case of classifying an object as an animal or not an animal—it's an either/or solution. There are just two possible outcome answers. This concept is typically represented as a 0 or a 1 in coding. Examples include:

i) Whether or not to lend to a bank customer (outcomes are yes or no).
ii) Assessing cancer risk (outcomes are high or low).
iii) Will a team win tomorrow's game (outcomes are yes or no).

2.Multinomial logistic regression: Multinomial logistic regression is a model where there are multiple classes that an item can be classified as. There is a set of three or more predefined classes set up prior to running the model. Examples include:

i) Classifying texts into what language they come from.
ii) Predicting whether a student will go to college, trade school or into the workforce.
iii) Does your cat prefer wet food, dry food or human food?



Types of Logistic Regression

Response Variable

Type of Logistic Regression

Two Categories

Binary

Yes No

Binary

1. 二项分类logistic回归

Three or More Categories

Nominal

Ordinal

Nominal

2. 多项分类logistic回归

Ordinal

3. 有序分类logistic回归

Figure 5.3: Logistic Regression Types

<u>3.Ordinal logistic regression:</u> Ordinal logistic regression is also a model where there are multiple classes that an item can be classified as; however, in this case an ordering of classes is required. Classes do not need to be proportionate. The distance between each class can vary. Examples include:

i) Ranking restaurants on a scale of 0 to 5 stars.
ii) Predicting the podium results of an Olympic event.
iii) Assessing a choice of candidates, specifically in places that institute ranked-choice voting.

5.2.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) is primarily a classier method that performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels. The support-vector clustering algorithm, created by HavaSiegelmann and Vladimir Vapnik, applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data, and is one of the most widely used clustering algorithms in industrial applications.

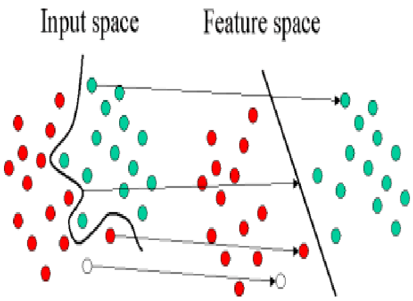


Figure 5.4(a): SVM Structure

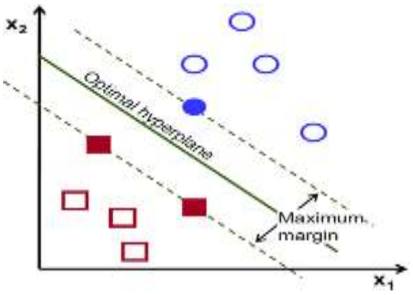


Figure 5.5(b): SVM Structure

SVM supports both regression and classification tasks and can handle multiple continuous and categorical variables. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall in addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. When data are unlabelled, supervised learning is not possible, and an unsupervised learning approach is required, which attempts to find natural clustering of the data to groups, and then map new data to these formed groups.

It is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyper-plane which categorizes new examples. A hyperplane is a subspace whose dimension is one less than that of its ambient space. If a space is 3-dimensional then its hyperplanes are the 2-dimensional planes, while if the space is 2-dimensional, its hyperplanes are the 1-dimensional lines. ... By its nature, it separates the space into two half spaces.

5.2.2.1 How Does SVM Work?

The basics of Support Vector Machines and how it works are best understood with a simple example. Let’s imagine we have two tags: red and blue, and our data has two features: x and y. We want a classifier that, given a pair of (x,y) coordinates, outputs if it’s either red or blue. We plot our already labeled training data on a plane:

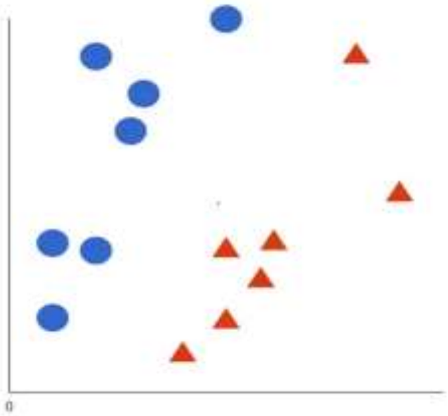


Figure 5.5(a): Labeled Data

A support vector machine takes these data points and outputs the hyperplane (which in two dimensions it’s simply a line) that best separates the tags. This line is the decision boundary: anything that falls to one side of it we will classify as blue, and anything that falls to the other as red.

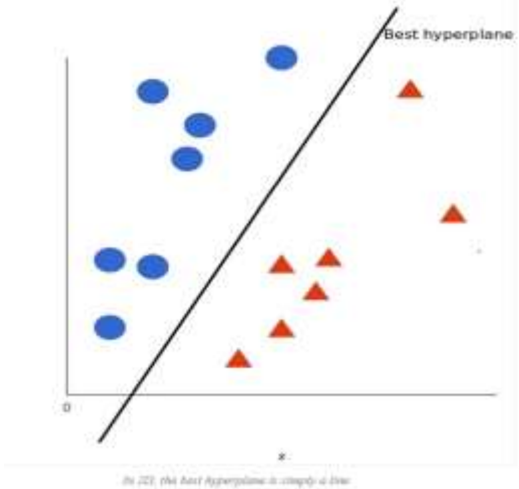


Figure 5.5(b): Divided data with Hyperplane

But, what exactly is the best hyperplane? For SVM, it's the one that maximizes the margins from both tags. In other words: the hyperplane (remember it's a line in this case) whose distance to the nearest element of each tag is the largest.

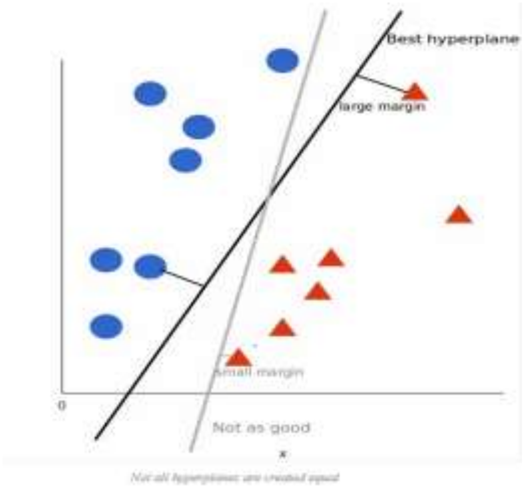


Figure 5.5(c): Divided Data with different Hyperplane

Nonlinear data: Now this example is easy, since clearly the data was linearly separable — we could draw a straight line to separate red and blue. Sadly, usually things aren't that simple. Let us take a look at this case:

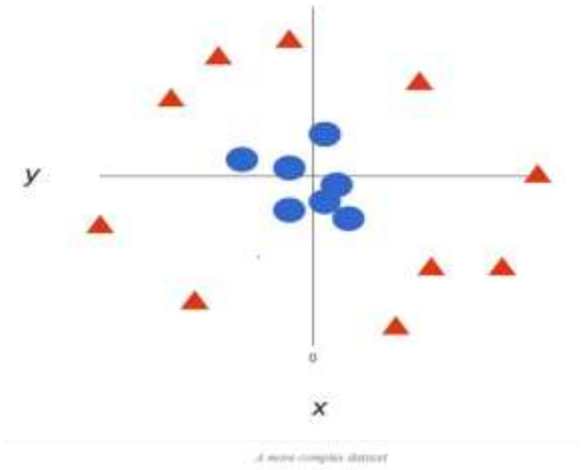
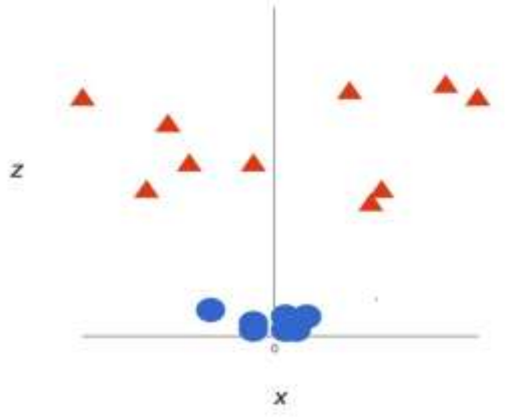


Figure 5.5(d): Linearly separable Data

It's pretty clear that there's not a linear decision boundary (a single straight line that separates both tags). However, the vectors are very clearly segregated and it looks as though it should be easy to separate them, so here's what we'll do: we will add a third dimension. Up until now we had two dimensions: x and y. We create a new z dimension, and we rule that it be calculated a certain way that is convenient for us: $z = x^2 + y^2$.This will give us a three-dimensional (3D) space [20].

Taking a slice of that space, it looks like the figure below:



From a different perspective, the data is now in two linearly separated groups

Figure 5.5(e): Data with two Linearly separated groups

The result is:

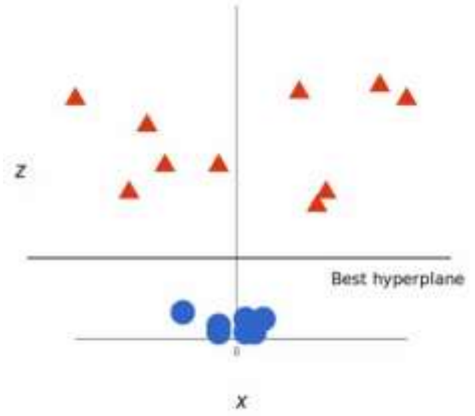
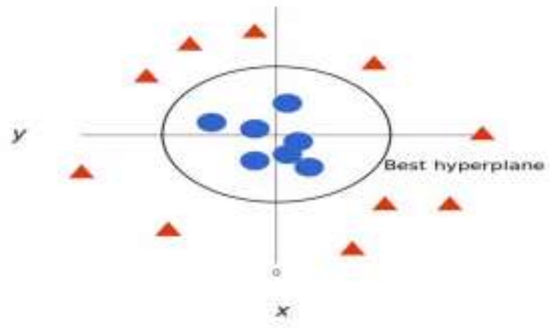


Figure 5.5(f): Hyperplane parallel to the x-axis

That’s great! Note that since we are in three dimensions now, the hyperplane is a plane parallel to the x axis at a certain z (let’s say z = 1).What’s left is mapping it back to two dimensions:



Back to our original view, everything is now neatly separated

Figure 5.5(g): Data with Best Hyperplane

5.2.2.2 Advantages of SVM

Advantages of support vector machine are:

i) Support vector machine works comparably well when there is an understandable margin of dissociation between classes.

ii) It is more productive in high dimensional spaces.

iii) It is effective in instances where the number of dimensions is larger than the number of specimens.

iv) Support vector machine is comparably memory systematic.

5.2.2.3 Applications of SVM

SVMs can be used to solve various real-world problems:

i) SVMs are helpful in text and hypertext categorization, as their application can significantly reduce the need for labeled training instances in both the standard inductive and transductive settings. Some methods for shallow semantic parsing are based on support vector machines.

ii) Classification of images can also be performed using SVMs. Experimental results show that SVMs achieve significantly higher search accuracy than traditional query refinement schemes after just three to four rounds of relevance feedback. This is also true for image segmentation systems, including those using a modified version SVM that uses the privileged approach as suggested by Vapnik.

iii) Hand-written characters can be recognized using SVM.

iv) The SVM algorithm has been widely applied in the biological and other sciences. They have been used to classify proteins with up to 90% of the compounds classified correctly. Permutation tests based on SVM weights have been suggested as a mechanism for interpretation of SVM models.Support-vector machine weights have also been used to interpret SVM models in the past. Postdoc interpretation of support-vector machine models in order to identify features used by the model to make predictions is a relatively new area of research with special significance in the biological sciences.

5.2.3 K-Nearest Neighbour (KNN)

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

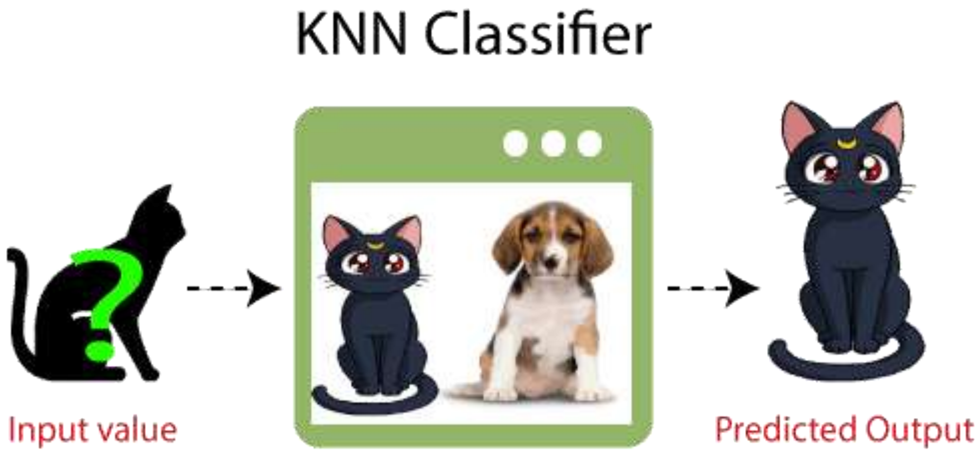


Figure 5.6: KNN Classifier

5.2.3.1 Why do we need a KNN Algorithm?

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

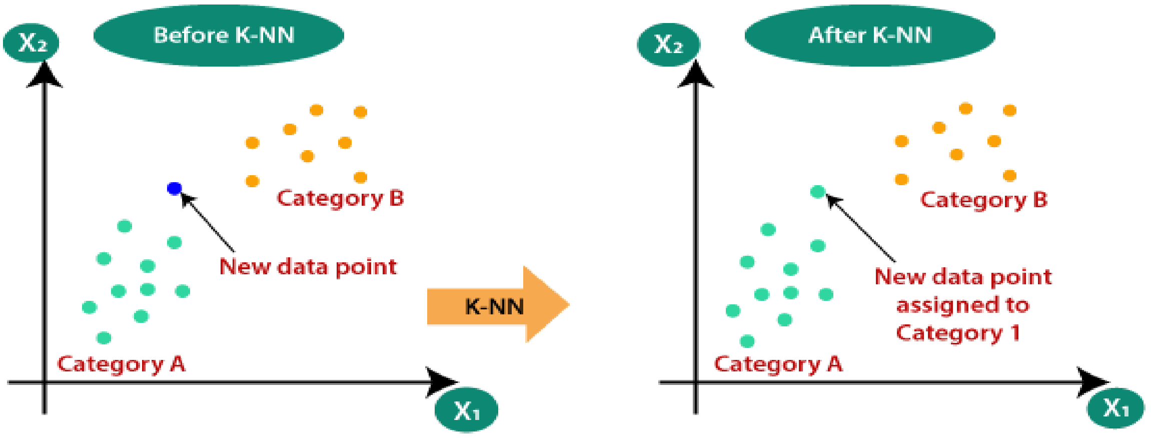


Figure 5.7: Apply KNN Algorithm

5.2.3.2 How Does KNN Work?

The K-NN working can be explained on the basis of the below algorithm:

- Step-1: Select the number K of the neighbors
- Step-2: Calculate the Euclidean distance of K number of neighbors
- Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.
- Step-4: Among these k neighbors, count the number of the data points in each category.
- Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.
- Step-6: Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:

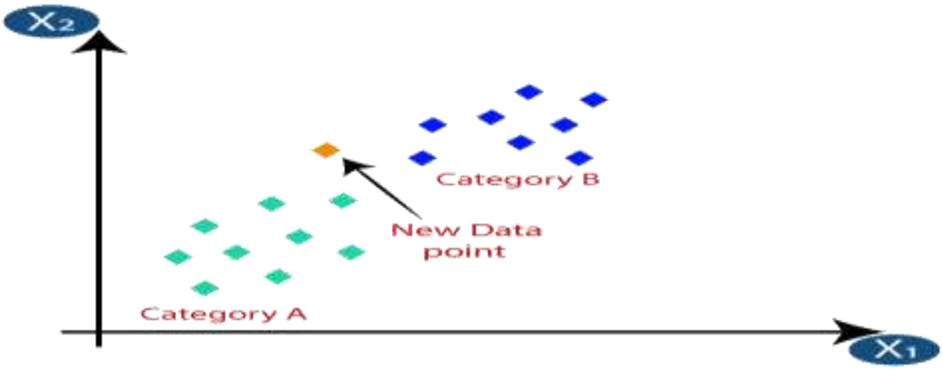


Figure 5.8(a): Data point

- Firstly, we will choose the number of neighbors, so we will choose the k=5.
- Next, we will calculate the Euclidean distance between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:

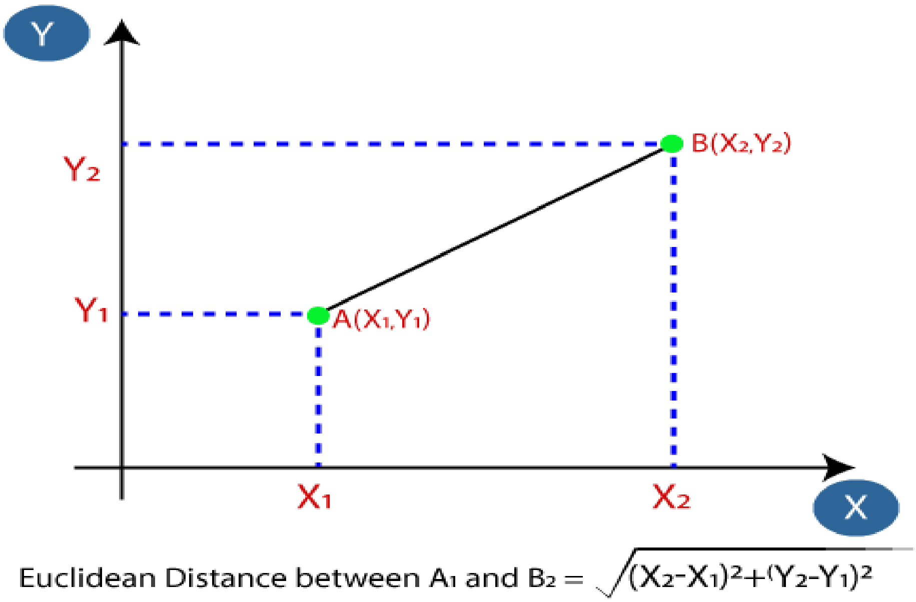

Y
X
Y_2
Y_1
$B(X_2,Y_2)$
$A(X_1,Y_1)$
X_1
X_2
Euclidean Distance between A_1 and $B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

Figure 5.8(b): Euclidean Distance

- By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:

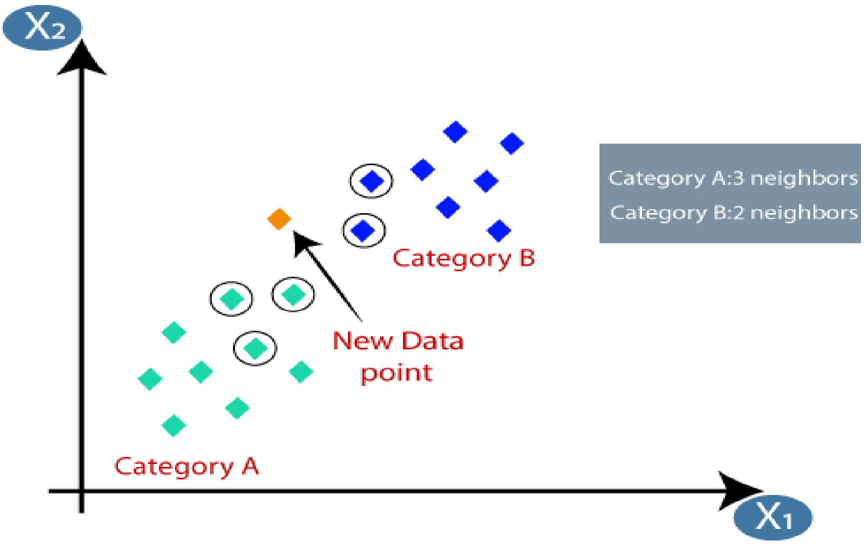


Figure 5.8(c): Data Category

- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

5.2.3.3Advantages of KNN

KNN is called Lazy Learner (Instance based learning). It does not learn anything in the training period. It does not derive any discriminative function from the training data. In other words, there is no training period for it. It stores the training dataset and learns from it only at the time of making real time predictions. This makes the KNN algorithm much faster than other algorithms that require training e.g. SVM, Linear Regression etc.

Since the KNN algorithm requires no training before making predictions, new data can be added seamlessly which will not impact the accuracy of the algorithm.

KNN is very easy to implement. There are only two parameters required to implement KNN i.e. the value of K and the distance function (e.g. Euclidean or Manhattan etc.)

5.2.3.4 Applications of KNN

The KNN algorithm has been utilized within a variety of applications, largely within classification. Some of these use cases include:

Data preprocessing: Datasets frequently have missing values, but the KNN algorithm can estimate for those values in a process known as missing data imputation.

Recommendation Engines: Using click stream data from websites, the KNN algorithm has been used to provide automatic recommendations to users on additional content. This research (link resides outside of ibm.com) shows that the a user is assigned to a particular group, and based on that group's user behavior, they are given a recommendation. However, given the scaling issues with KNN, this approach may not be optimal for larger datasets.

Finance: It has also been used in a variety of finance and economic use cases. For example, one paper (PDF, 391 KB) (link resides outside of ibm.com) shows how using KNN on credit data can help banks assess risk of a loan to an organization or individual. It is used to determine the credit-worthiness of a loan applicant. Another journal (PDF, 447 KB)(link resides outside of ibm.com) highlights its use in stock market forecasting, currency exchange rates, trading futures, and money laundering analyses.

Healthcare: KNN has also had application within the healthcare industry, making predictions on the risk of heart attacks and prostate cancer. The algorithm works by calculating the most likely gene expressions.

Pattern Recognition: KNN has also assisted in identifying patterns, such as in text and digit classification (link resides outside of ibm.com). This has been particularly helpful in identifying handwritten numbers that you might find on forms or mailing envelopes.

5.2.4 Naive Bayes (NB)

Naive bayes in machine learning is defined as probabilistic model in machine learning technique in the genre of supervised learning that is used in varied use cases of mostly classification, but applicable to regression (by force fit of-course!) as well. The reason of putting a naive in front of the algorithm name is because it assumes that the features that goes into the model are independent of each other or in other words, changes done to one variable doesn’t affect any others. Though this assumption is a strong one and happens to be a basic one and, the effect of the variable is present. Due to this algorithm, Naive Bayes happens to be a simple, yet very powerful algorithm. Due to its lesser complexity this is the go-to choice for any of the algorithms where in one either has to respond to a request quickly or one needs to perform some calculation in order to provide some basic yet powerful insights from the data!

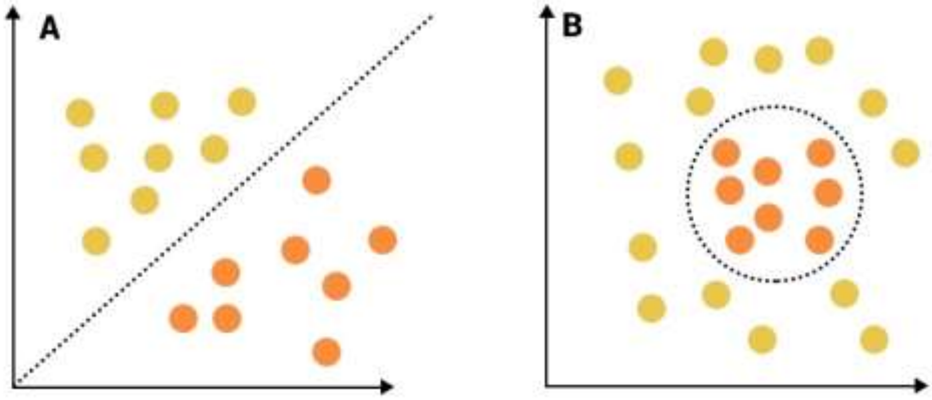


Figure 5.9:Naive Bayes

5.2.4.2 How Does NB Work?

In the introduction we have understood that Naive Bayes is a simple algorithm that assumes the factors that are a part of the analysis are independent to each other and that the answer to Naive in its name. but what about the other word Bayes. In order to have a deeper understanding of Naive Bayes algorithm, we would first understand the Bayes theorem in conditional probability.

Let us understand the Naive Bayes with an example. Let us take the very generic example, rolling a dice. What is the probability that we would get a number 2? Since there is one “2” in the dice, and total 6 possibilities, the probability is 1/6. Now, let us enhance the example a bit more. We are at a guessing game, where one has to ask some questions and on the basis of that, we have to guess the number and pattern on the card. At first, we would have 4 possibilities and only one is true, and hence the probability is ¼. Let say we have guessed the color of the pattern, now to guess the pattern there are only 2 possibilities present and one of them has to be true and hence the probability of us being correct is ½ which is obviously more than ¼. Did you notice the difference in the statements? In the case of dice, there is no condition, whereas in case of cards the condition is that the color is already known. Thus, probability of an event to occur when a condition is given is known and conditional probability and the theorem that dictates this is known as Bayes theorem.

The mathematical equation of the same is given as:

P(Y|X)=(P(X|Y)*P(Y))/P(X)

And it is read as probability of Y to happen given that X has happened is equal to probability of X to happen such that Y has occurred multiplied with probability of Y to occur upon the probability of X to happen. Here, P(Y|X) is known as Posterior probability, P(X|Y) is known s a likelihood probability, P(Y) is termed as prior probability and P(X) is the probability of evidence.

Now keeping the above theorem in mind, let us see the working of Naive Bayes. In real-world examples, we might have multiple X variables in the data and with the statement that multiple X variables are independent we can substantiate that the probability will follow a multiplicative relation to finding the probability. The equation will look something like:

P(Y/(X_1,X_2,X_3,X_4…X_n,))=(P(X_1 | Y)*P(X_2 | Y)*P(X_3 | Y)*…*P(X_n | Y))/(P(X_1)*P(X_2)*P(X_3)*…*P(X_n))

With setting of this context, now let us take an example to fully understand the working of Naive Bayes through an example in classification. The working takes place in 3 steps. Let us look at the steps first and then take one example and get a hands-on understanding of the working.

At first, the frequency table is built on the basis of categories in the features. This frequency table is based on the category of the features that go into the model. For example, a column has 3 categories; all these categories are taken, and corresponding frequency of occurrence is tabulated. In the next step, the corresponding probabilities are calculated against the categories and their corresponding frequency of the target variable. Finally, the above calculation using the formula of Bayes theorem is used to calculate the posterior probability when any new dataset is sent for prediction.

It is now time to look at the steps with an example. Let us see if a person is likely to buy a gadget given the income status:

Original Data: (Table: 5.1)

Income Status	**Gadget Bought**
Low	Yes
Mid	Yes
High	Yes
High	Yes
Mid	No
Low	No
Mid	Yes
High	Yes
Low	No
Mid	No
Mid	Yes
Low	No
High	Yes
High	Yes

Step 1: Frequency table creation **(Table: 5.2)**

Income Group	**Yes**	**No**
High	5	0
Mid	3	2
Low	1	3
Total	9	5

Step 2: Probability calculation of the likelihood income groups **(Table: 5.3)**

Income Group	**Yes**	**No**	
High	5	0	5/14 = 0.35
Mid	3	2	5/14 = 0.35
Low	1	3	4/14 = 0.3
Total	9/14 = 0.64	5/14 = 0.36	

Step 3: Now whenever a new dataset comes, we can tell the corresponding probability that whether the person will buy the gadget or not by using the posterior probability formulae. Let us say that the new data is with income group as Mid. We would need to find out if the person will buy a gadget or not.

P(Yes|Mid) = P(Mid|Yes)*P(Yes)/P(Mid)

Here,

P(Mid|Yes) = 3/9 [Total frequency of Mid in Yes = 3, and total frequency in Yes = 9]
P(Yes) = 9/14
P(Mid) = 5/14

Substituting the values, we get P(Yes|Mid) = (3/9*9/14)/(5/14) = 3/5

Similarly, for P(No|Mid) = 2/5

So, we can conclude that if the person is of Income Category Mid, he has a higher propensity to buy the gadget.

5.2.4.3 Advantages of NB

The following are some of the benefits of the Naive Bayes classifier:

It is simple and easy to implement

It doesn’t require as much training data

It handles both continuous and discrete data

It is highly scalable with the number of predictors and data points

It is fast and can be used to make real-time predictions

It is not sensitive to irrelevant features

5.2.3.4 Applications of NB

Here are some areas where this algorithm finds applications:

Text Classification: Most of the time, Naive Bayes finds uses in-text classification due to its assumption of independence and high performance in solving multi-class problems. It enjoys a high rate of success than other algorithms due to its speed and efficiency.

Sentiment Analysis: One of the most prominent areas of machine learning is sentiment analysis, and this algorithm is quite useful there as well. Sentiment analysis focuses on identifying whether the customers think positively or negatively about a certain topic (product or service).

Recommender Systems: With the help of Collaborative Filtering, Naive Bayes Classifier builds a powerful recommender system to predict if a user would like a particular product (or resource) or not. Amazon, Netflix, and Flipkart are prominent companies that use recommender systems to suggest products to their customers.

CHAPTER 6

EXPERIMENTS AND RESULTS

6.1 Introduction

Crime data analysis is a systematic analysis for detecting and analyzing various types of crimes and classifies its patterns. These patterns play an important role for solving different crime types, problems and in making different strategies to solve the crime problems. Different types of news articles of online newspapers publish thousands of crime news which contain the details of victims, crime type, criminals, locations etc. Many studies have discovered various techniques to investigate the crime data. Now various machine learning techniques, statistical methods, deep learning approaches are utilized for the extraction of appropriate meaningful features in order to classify the text documents. In the active research area of text mining, Text categorization is one where all the documents are categorized with basic three types of knowledge, supervised, semi-supervised and unsupervised.

Among these machine learning techniques, supervised learning has become more popular in the word. Different types of supervised learning approaches have been used in many researches such as Logistic Regression, Support Vector Machine, K-Nearest Neighbour, Naive Bayes, etc. In order to train these supervised learning model, different statistical and machine learning approaches have been used to extract meaningful features for accurate text classifications.

Large datasets on different language are available, and many researches on text classification have also been done in different languages. In the field of Bangla text categorization, only a couple of works have been done. A few of them work on large dataset and these works have been done on general categorization of textual document.

6.2 Dataset of Crime News Articles

We presented our proposed dataset about different types of crimes happened in Bangladesh. We fetched these crime articles from different Bangla newspapers. To do this we built a web-spider that will crawl online news portals and scrap news for our application. This can be done using “Scrapy”, which is a popular python framework for scraping data. By using this we fetched different types of crime news articles from renowned newspapers in Bangladesh. We collected crime news from following online portals:

- Prothom Alo (https://www.prothomalo.com/)
- Jugantor (https://www.jugantor.com/)

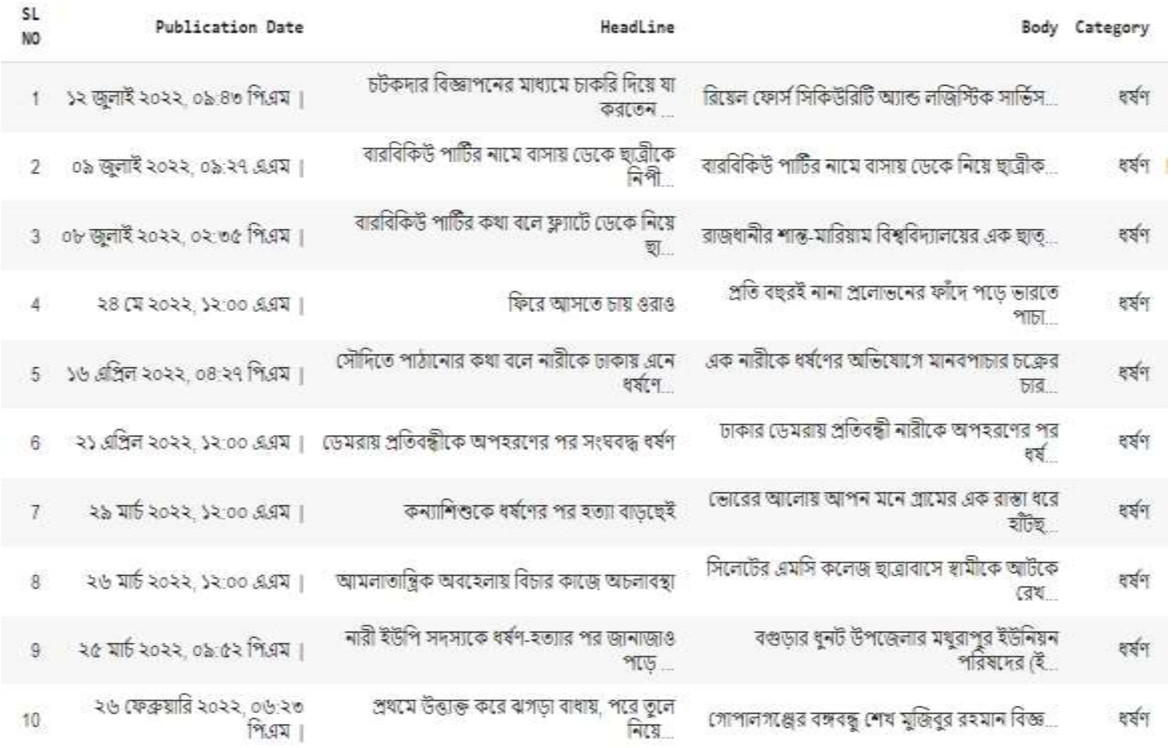
- Noya Digonto (www.dailynayadiganta.com/)

We considered the crime news which belonged to nine pre-defined classes: MURDER, RAPE, ROBBERY, DRUGS, THEFT, CORRUPTION, FRAUD, ABDUCTION and OTHERS (non crime news like entertainment, sports, etc). We labeled the classes manually by reading each of the news articles. We applied different filters to avoid duplicate Crime News. The distribution of the total 3,500 Crime articles on nine categories are shown on the table:

Table 6.1: Categories of crimes and basic statistics-

Category	**No. of Crime News**	**No. of Total Sentence**	**No. of Total Word**
খুন (MURDER)	500	18347	194150
ধর্ষণ (RAPE)	500	19231	166772
অপহরণ (ABDUCTION)	200	6135	67548
ছিনতাই (THEFT)	300	11121	114522
ডাকাতি (ROBBERY)	200	7535	100621
মাদক (DRUGS)	400	5894	74709
দুর্নীতি (CORRUPTION)	301	14601	156011
প্রতারণা (FRAUD)	399	15559	152174
অন্যান্য (OTHERS)	700	21300	271068

And the example of our dataset is given in the figure 6.1:



SL NO	Publication Date	HeadLine	Body	Category
1	১২ জুলাই ২০২২, ০৯:৪৩ পিএম |	চটকদার বিজ্ঞাপনের মাধ্যমে চাকরি দিয়ে যা করতেন ...	রিয়েল ফোর্স সিকিউরিটি অ্যান্ড লজিস্টিক সার্ভিস...	ধর্ষণ
2	০৯ জুলাই ২০২২, ০৯:২৭ এএম |	বারবিকিউ পার্টির নামে বাসায় ডেকে ছাত্রীকে নিপী...	বারবিকিউ পার্টির নামে বাসায় ডেকে নিয়ে ছাত্রীক...	ধর্ষণ
3	০৮ জুলাই ২০২২, ০২:৩৫ পিএম |	বারবিকিউ পার্টির কথা বলে ফ্ল্যাটে ডেকে নিয়ে ছা...	রাজধানীর শান্ত-মারিয়াম বিশ্ববিদ্যালয়ের এক ছাত্...	ধর্ষণ
4	২৪ মে ২০২২, ১২:০০ এএম |	ফিরে আসতে চায় ওরাও	প্রতি বছরই নানা প্রলোভনের ফাঁদে পড়ে ভারতে পাচা...	ধর্ষণ
5	১৬ এপ্রিল ২০২২, ০৪:২৭ পিএম |	সৌদিতে পাঠানোর কথা বলে নারীকে ঢাকায় এনে ধর্ষণে...	এক নারীকে ধর্ষণের অভিযোগে মানবপাচার চক্রের চার...	ধর্ষণ
6	২১ এপ্রিল ২০২২, ১২:০০ এএম |	ডেমরায় প্রতিবন্ধীকে অপহরণের পর সংঘবদ্ধ ধর্ষণ	ঢাকার ডেমরায় প্রতিবন্ধী নারীকে অপহরণের পর ধর্ষ...	ধর্ষণ
7	২৯ মার্চ ২০২২, ১২:০০ এএম |	কন্যাশিশুকে ধর্ষণের পর হত্যা বাড়ছেই	ভোরের আলোয় আপন মনে গ্রামের এক রাস্তা ধরে হাঁটছি...	ধর্ষণ
8	২৬ মার্চ ২০২২, ১২:০০ এএম |	আমলাতান্ত্রিক অবহেলায় বিচার কাজে অচলাবস্থা	সিলেটের এমসি কলেজ ছাত্রাবাসে স্বামীকে আটকে রেখ...	ধর্ষণ
9	২৫ মার্চ ২০২২, ০৯:৫২ পিএম |	নারী ইউপি সদস্যকে ধর্ষণ-হত্যার পর জানাজাও পড়ে ...	বগুড়ার ধুনট উপজেলার মথুরাপুর ইউনিয়ন পরিষদের (ই...	ধর্ষণ
10	২৬ ফেব্রুয়ারি ২০২২, ০৬:২৩ পিএম |	প্রথমে উত্ত্যক্ত করে ঝগড়া বাধায়, পরে তুলে নিয়ে...	গোপালগঞ্জের বঙ্গবন্ধু শেখ মুজিবুর রহমান বিজ্ঞ...	ধর্ষণ

Figure 6.1: A snapshot of our own dataset

6.3 Proposed Model

First, we will have a news crawler that will crawl news articles from different online news portals. We need to classify whether news is a crime related article or not. It will also classify the type of crime. However we need to build the classifier model for the purpose. So, we developed Bangla Crime classifier by utilizing our dataset. We used machine learning algorithms SVM, Logistic Regression, K-Nearest Neighbour, Naive Bayes in our dataset. Our model first conducted some preprocessing on the news contents such as "Pickle". Pickling is a way to convert a python object (list, dict, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script. In the training phase, at first, we divided the processed dataset into training set and testing set. we employed the training dataset to train models with classic machine learning algorithms. The proposed model of our classifier is given in figure.

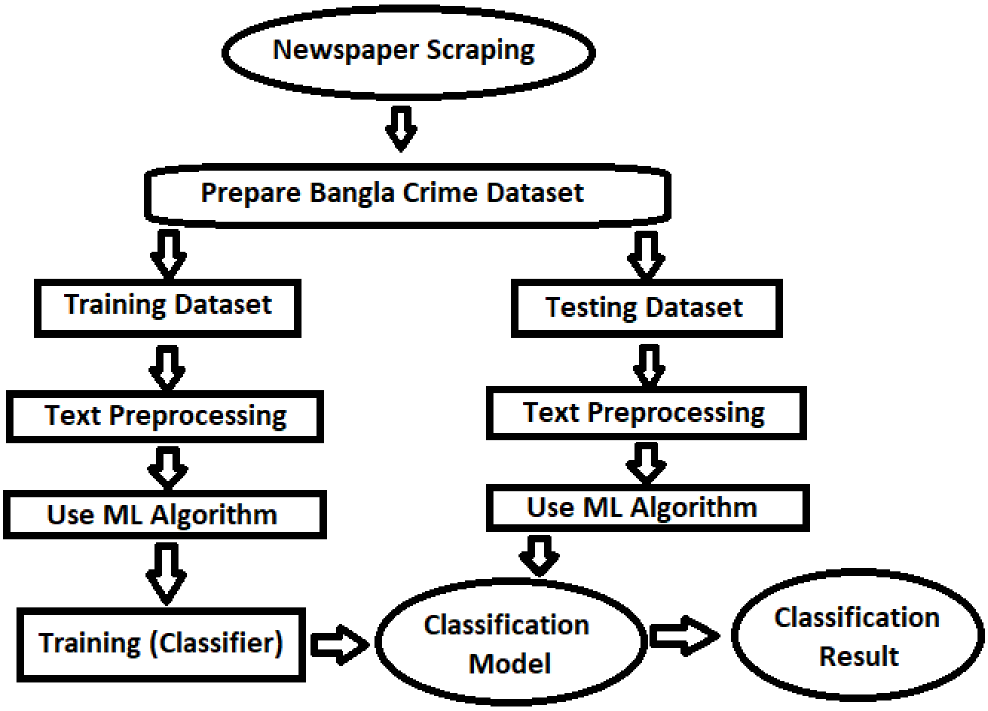


Figure 6.2: Proposed Algorithm of Classification process for Bangla Crime News

6.4 Experimental Result

We used state of the art machine learning algorithms (Logistic Regressing, Support Vector Machine, K-Nearest Neighbour, Naive Bayes) in training part. The total number of words and the total number of sentences in our dataset are given displayed in table 6.1. We calculated accuracy, precision, recall and F1 score by validating with our test data for all machine learning algorithms used, which are displayed in table 6.2.

6.4.1 Performance Metrics

To assess the performance of the prediction model we utilized Precision, Recall and F1 score as evaluation metrics. Beside accuracy, the precision and recall measures are also widely used in classification. Precision can be thought of as a measure of exactness (i.e., what percentage of tuples labeled as positive are actually such), whereas recall is a measure of completeness (what percentage of positive tuples are labeled as such). If recall is the same as sensitivity (or the true positive rate). These measures can be computed as,

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Here, TP = True Positives, FP = False Positives and FN =False Negative.

The F measure is the harmonic mean of precision and recall. It gives equal weight to precision and recall.

$$F1 - score = \frac{2 * (Recall * Precision)}{Recall + Precision}$$

6.4.2 Performance Analysis

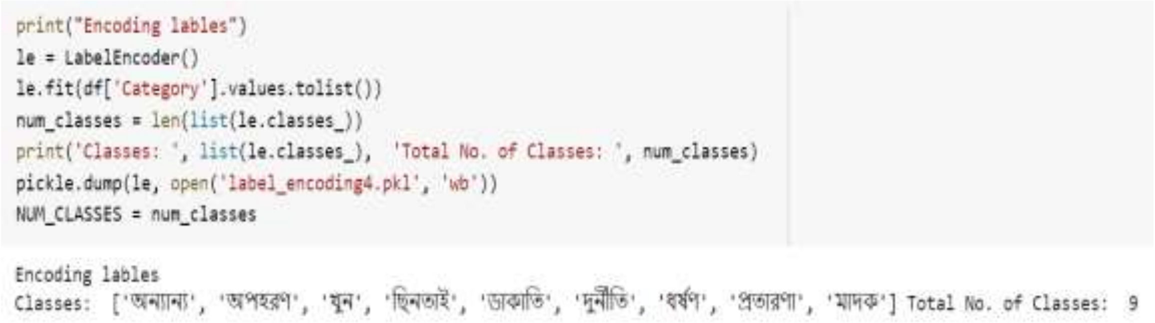
Table 6.2: State of the art Machine Learning

Learning Algorithm	**Recall**	**Precision**	**F1 Score**
Logistic Regression	0.8685	0.8688	0.8674
Support Vector Machine (SVM)	0.7952	0.8311	0.8016
K-Nearest Neighbour (KNN)	0.6504	0.6615	0.6442
Naive Bayes (NB)	0.8171	0.8202	0.8172

Table 6.2 shows the performance in four performance evaluation metrics: Accuracy, Precision, Recall and F1 Score using state-of-art machine learning algorithms such as Logistic Regression, SVM, KNN and Naive Bayes. Overall, Logistic Regression is the best performer among the three classic machine learning algorithm and the accuracy is around 87% for all four performance evaluation metrics. Naive Bayes is the 2nd best performer among the three classic machine learning algorithm and the accuracy is 81% for all four performance evaluation metrics. Naive Bayes is the 3rd best performer among the three classic machine learning algorithm and the accuracy is around 80% for all four

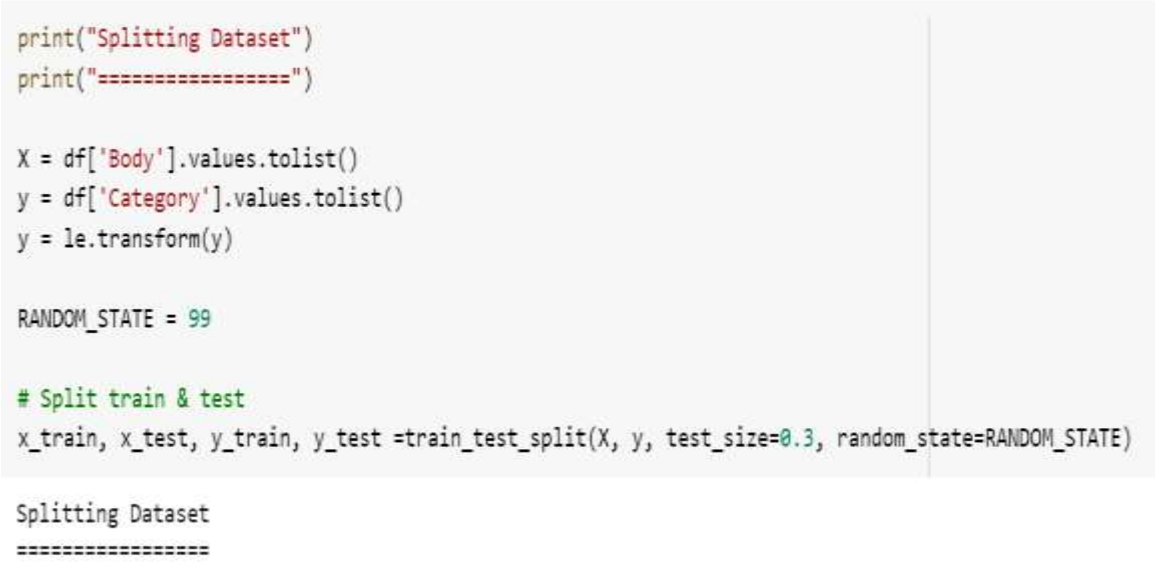
performance evaluation metrics. For KNN the performance is below 70% for both word vectors except KNN in all four performance evaluation metrics.

6.5 Empirical Data Analysis & Visualization of Result



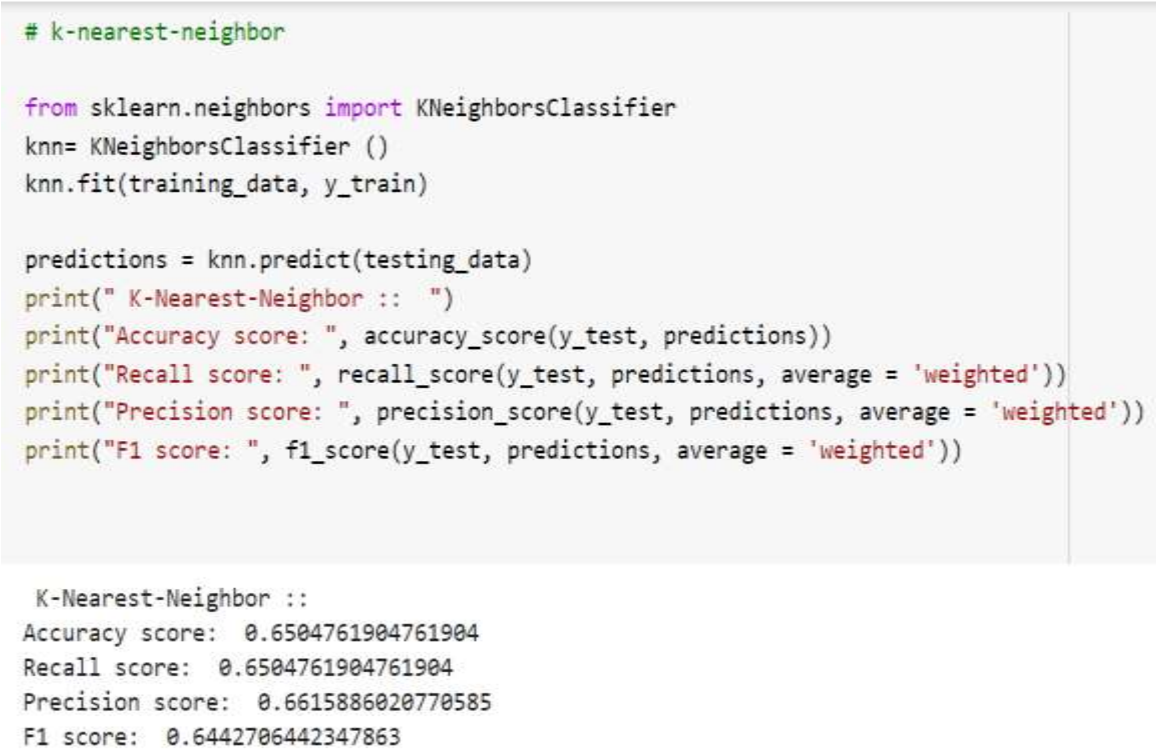
```
print("Encoding lables")
le = LabelEncoder()
le.fit(df['Category'].values.tolist())
num_classes = len(list(le.classes_))
print('Classes: ', list(le.classes_),  'Total No. of Classes: ', num_classes)
pickle.dump(le, open('label_encoding4.pkl', 'wb'))
NUM_CLASSES = num_classes
```

```
Encoding lables
Classes:  ['অন্যান্য', 'অপহরণ', 'খুন', 'ছিনতাই', 'ডাকাতি', 'দুর্নীতি', 'ধর্ষণ', 'প্রতারণা', 'মাদক'] Total No. of Classes:  9
```



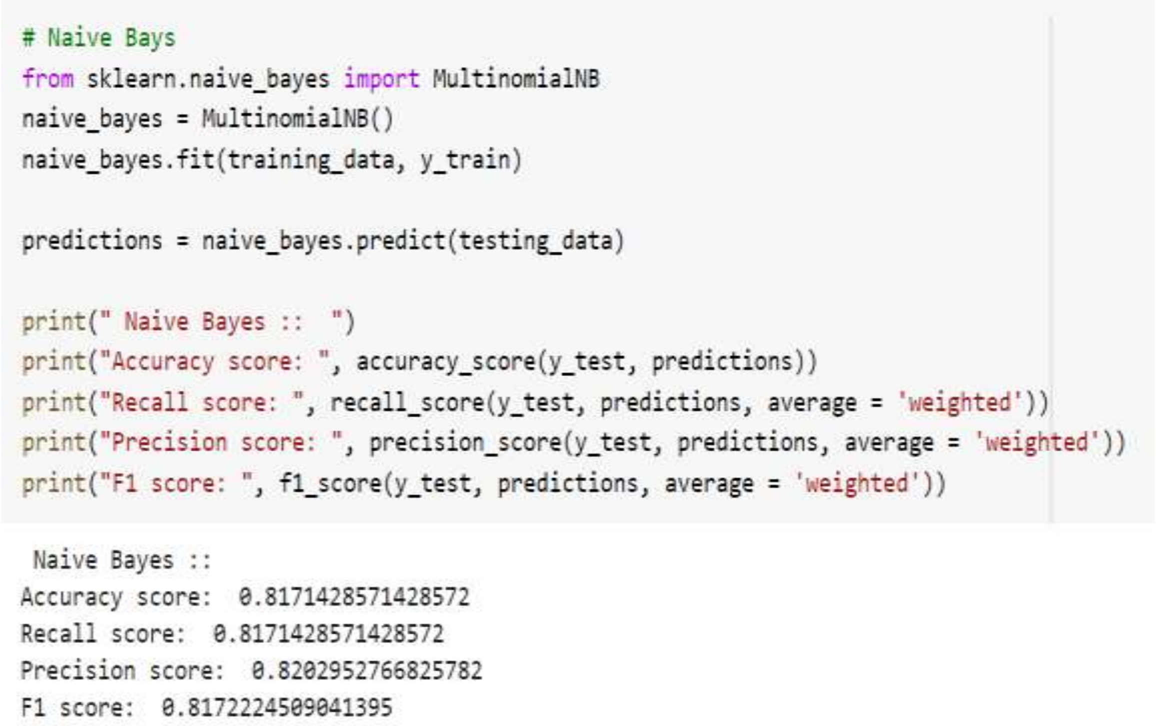
```
print("Splitting Dataset")
print("=================")

X = df['Body'].values.tolist()
y = df['Category'].values.tolist()
y = le.transform(y)

RANDOM_STATE = 99

# Split train & test
x_train, x_test, y_train, y_test =train_test_split(X, y, test_size=0.3, random_state=RANDOM_STATE)
```

```
Splitting Dataset
=================
```



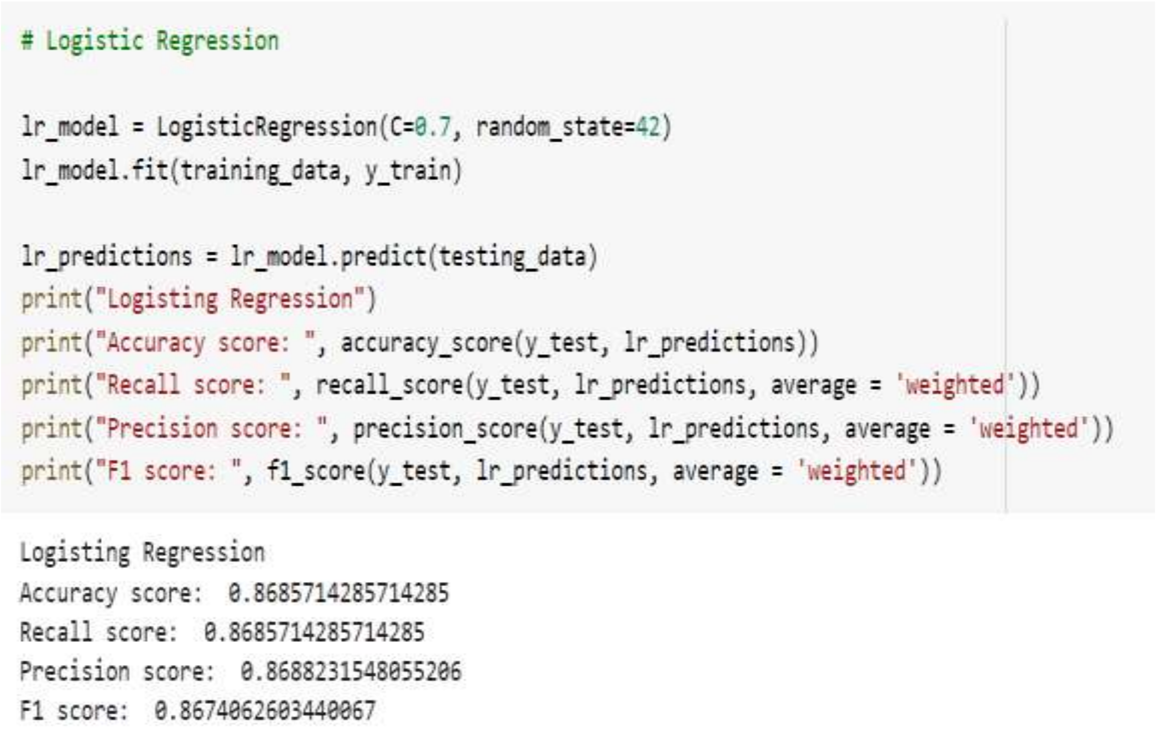
```
# k-nearest-neighbor

from sklearn.neighbors import KNeighborsClassifier
knn= KNeighborsClassifier ()
knn.fit(training_data, y_train)

predictions = knn.predict(testing_data)
print(" K-Nearest-Neighbor ::  ")
print("Accuracy score: ", accuracy_score(y_test, predictions))
print("Recall score: ", recall_score(y_test, predictions, average = 'weighted'))
print("Precision score: ", precision_score(y_test, predictions, average = 'weighted'))
print("F1 score: ", f1_score(y_test, predictions, average = 'weighted'))
```

```
 K-Nearest-Neighbor ::
Accuracy score:  0.6504761904761904
Recall score:  0.6504761904761904
Precision score:  0.6615886020770585
F1 score:  0.6442706442347863
```



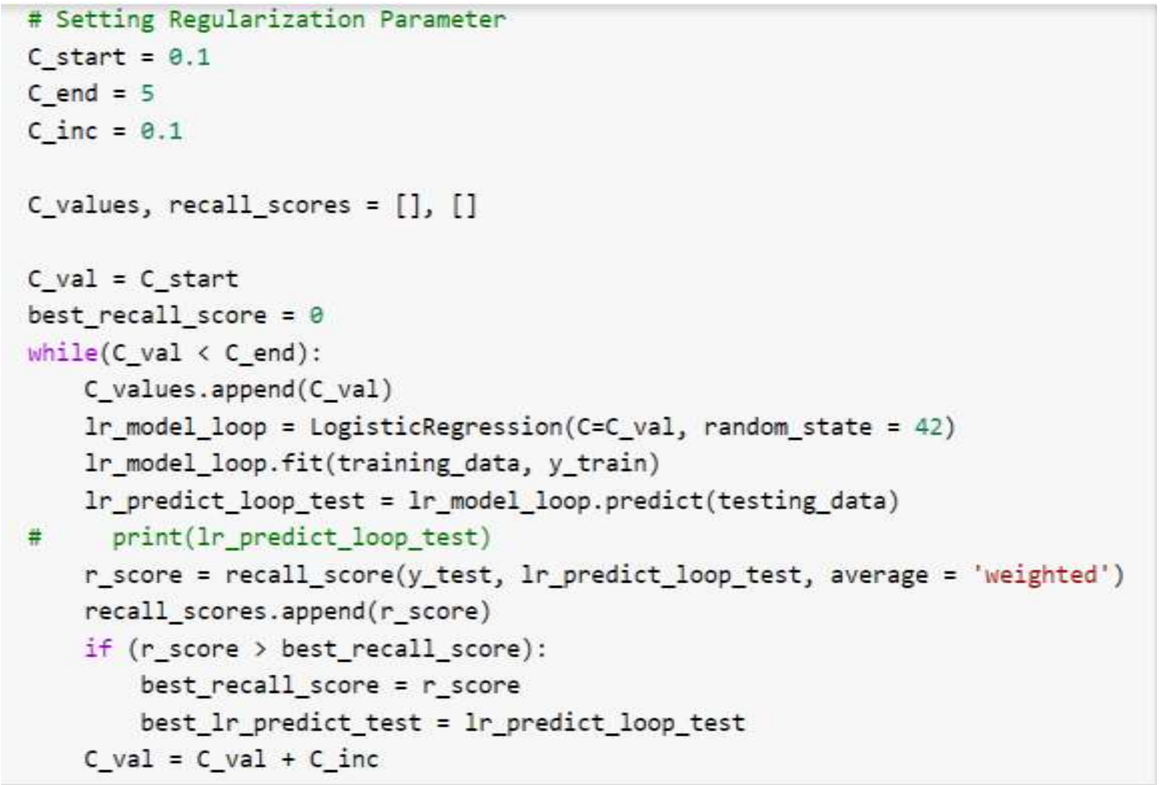
```
# Naive Bays
from sklearn.naive_bayes import MultinomialNB
naive_bayes = MultinomialNB()
naive_bayes.fit(training_data, y_train)

predictions = naive_bayes.predict(testing_data)

print(" Naive Bayes ::  ")
print("Accuracy score: ", accuracy_score(y_test, predictions))
print("Recall score: ", recall_score(y_test, predictions, average = 'weighted'))
print("Precision score: ", precision_score(y_test, predictions, average = 'weighted'))
print("F1 score: ", f1_score(y_test, predictions, average = 'weighted'))
```

```
 Naive Bayes ::
Accuracy score:  0.8171428571428572
Recall score:  0.8171428571428572
Precision score:  0.8202952766825782
F1 score:  0.8172224509041395
```

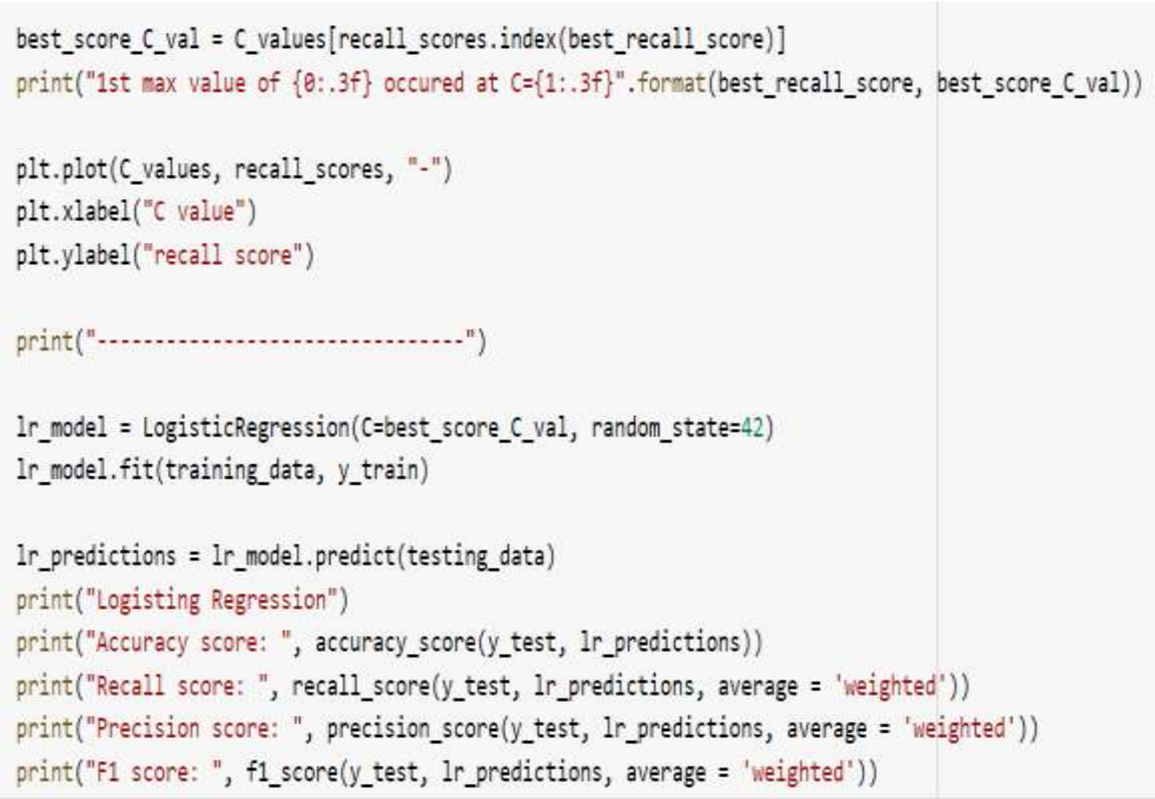


```
# Logistic Regression

lr_model = LogisticRegression(C=0.7, random_state=42)
lr_model.fit(training_data, y_train)

lr_predictions = lr_model.predict(testing_data)
print("Logisting Regression")
print("Accuracy score: ", accuracy_score(y_test, lr_predictions))
print("Recall score: ", recall_score(y_test, lr_predictions, average = 'weighted'))
print("Precision score: ", precision_score(y_test, lr_predictions, average = 'weighted'))
print("F1 score: ", f1_score(y_test, lr_predictions, average = 'weighted'))
```

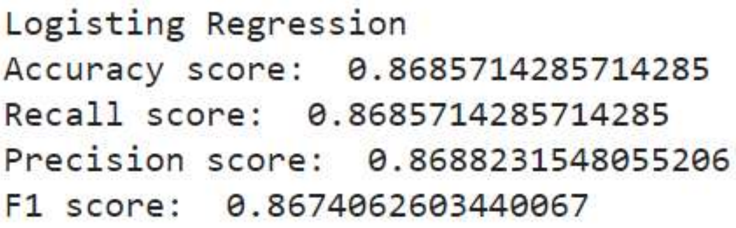
```
Logisting Regression
Accuracy score:  0.8685714285714285
Recall score:  0.8685714285714285
Precision score:  0.8688231548055206
F1 score:  0.8674062603440067
```



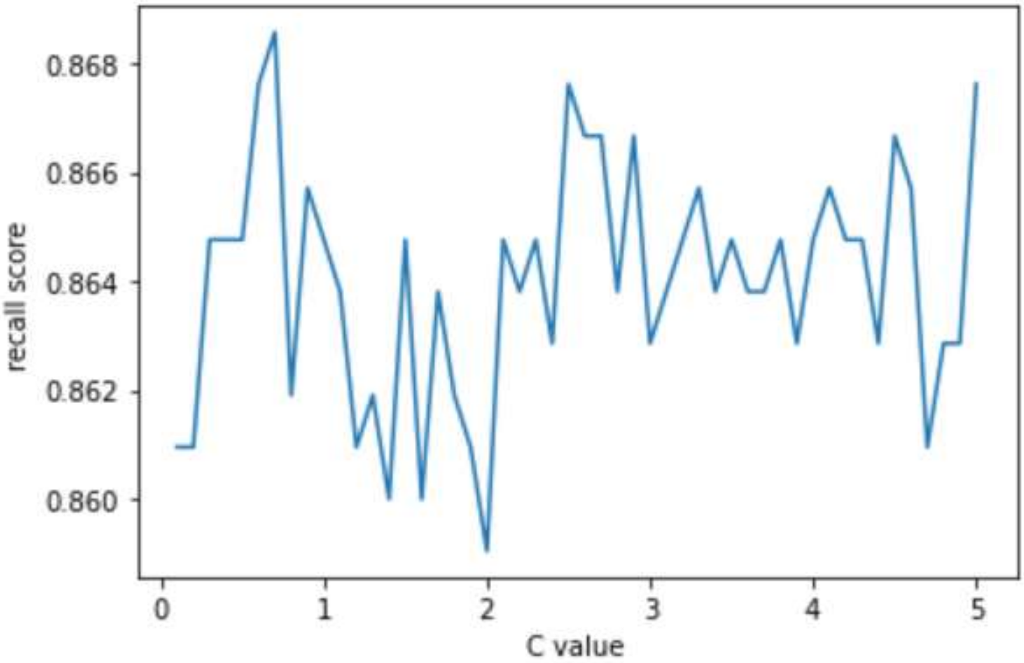
```
# Setting Regularization Parameter
C_start = 0.1
C_end = 5
C_inc = 0.1

C_values, recall_scores = [], []

C_val = C_start
best_recall_score = 0
while(C_val < C_end):
    C_values.append(C_val)
    lr_model_loop = LogisticRegression(C=C_val, random_state = 42)
    lr_model_loop.fit(training_data, y_train)
    lr_predict_loop_test = lr_model_loop.predict(testing_data)
#     print(lr_predict_loop_test)
    r_score = recall_score(y_test, lr_predict_loop_test, average = 'weighted')
    recall_scores.append(r_score)
    if (r_score > best_recall_score):
        best_recall_score = r_score
        best_lr_predict_test = lr_predict_loop_test
    C_val = C_val + C_inc
```

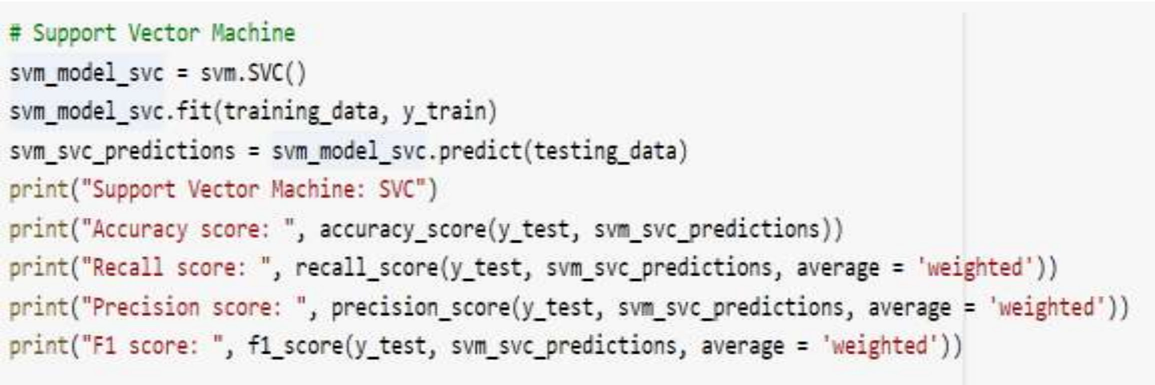


```
best_score_C_val = C_values[recall_scores.index(best_recall_score)]
print("1st max value of {0:.3f} occured at C={1:.3f}".format(best_recall_score, best_score_C_val))

plt.plot(C_values, recall_scores, "-")
plt.xlabel("C value")
plt.ylabel("recall score")

print("------------------------------")

lr_model = LogisticRegression(C=best_score_C_val, random_state=42)
lr_model.fit(training_data, y_train)

lr_predictions = lr_model.predict(testing_data)
print("Logisting Regression")
print("Accuracy score: ", accuracy_score(y_test, lr_predictions))
print("Recall score: ", recall_score(y_test, lr_predictions, average = 'weighted'))
print("Precision score: ", precision_score(y_test, lr_predictions, average = 'weighted'))
print("F1 score: ", f1_score(y_test, lr_predictions, average = 'weighted'))
```

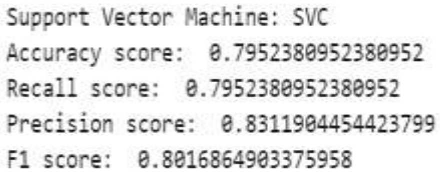


```
Logisting Regression
Accuracy score:  0.8685714285714285
Recall score:  0.8685714285714285
Precision score:  0.8688231548055206
F1 score:  0.8674062603440067
```

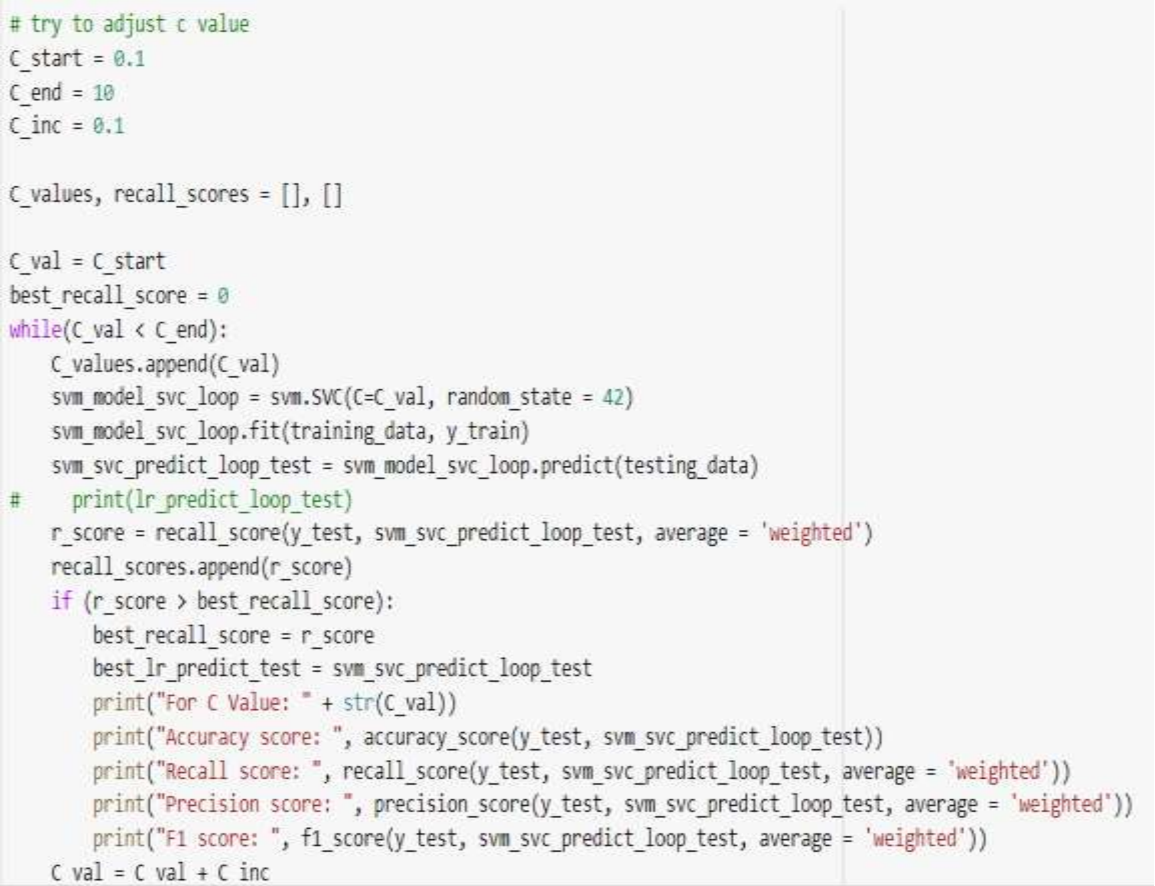




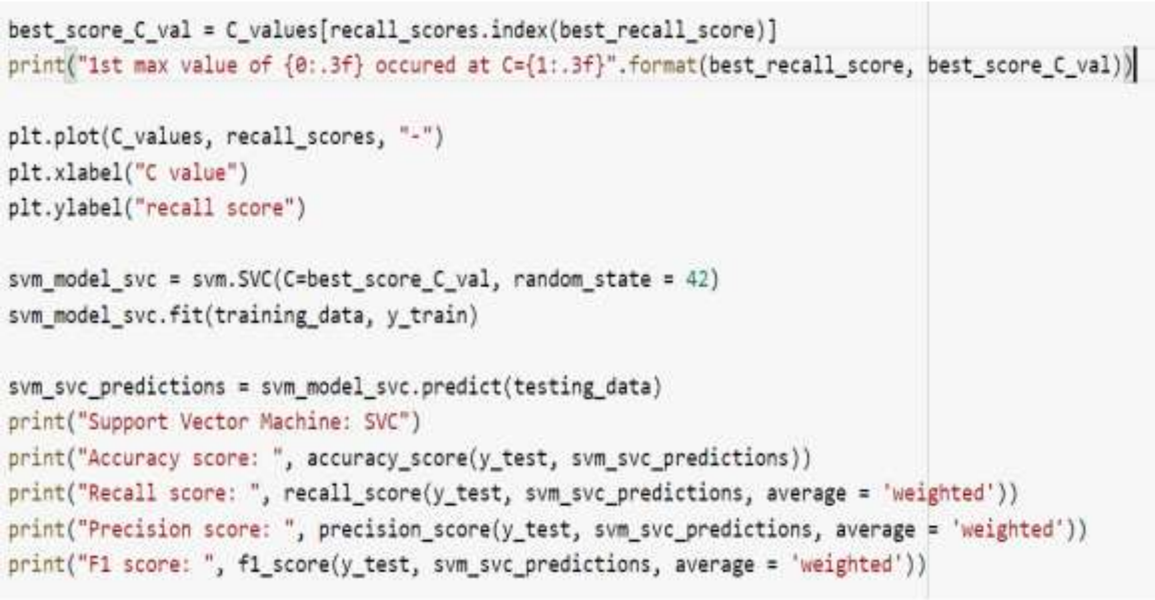
```
# Support Vector Machine
svm_model_svc = svm.SVC()
svm_model_svc.fit(training_data, y_train)
svm_svc_predictions = svm_model_svc.predict(testing_data)
print("Support Vector Machine: SVC")
print("Accuracy score: ", accuracy_score(y_test, svm_svc_predictions))
print("Recall score: ", recall_score(y_test, svm_svc_predictions, average = 'weighted'))
print("Precision score: ", precision_score(y_test, svm_svc_predictions, average = 'weighted'))
print("F1 score: ", f1_score(y_test, svm_svc_predictions, average = 'weighted'))
```



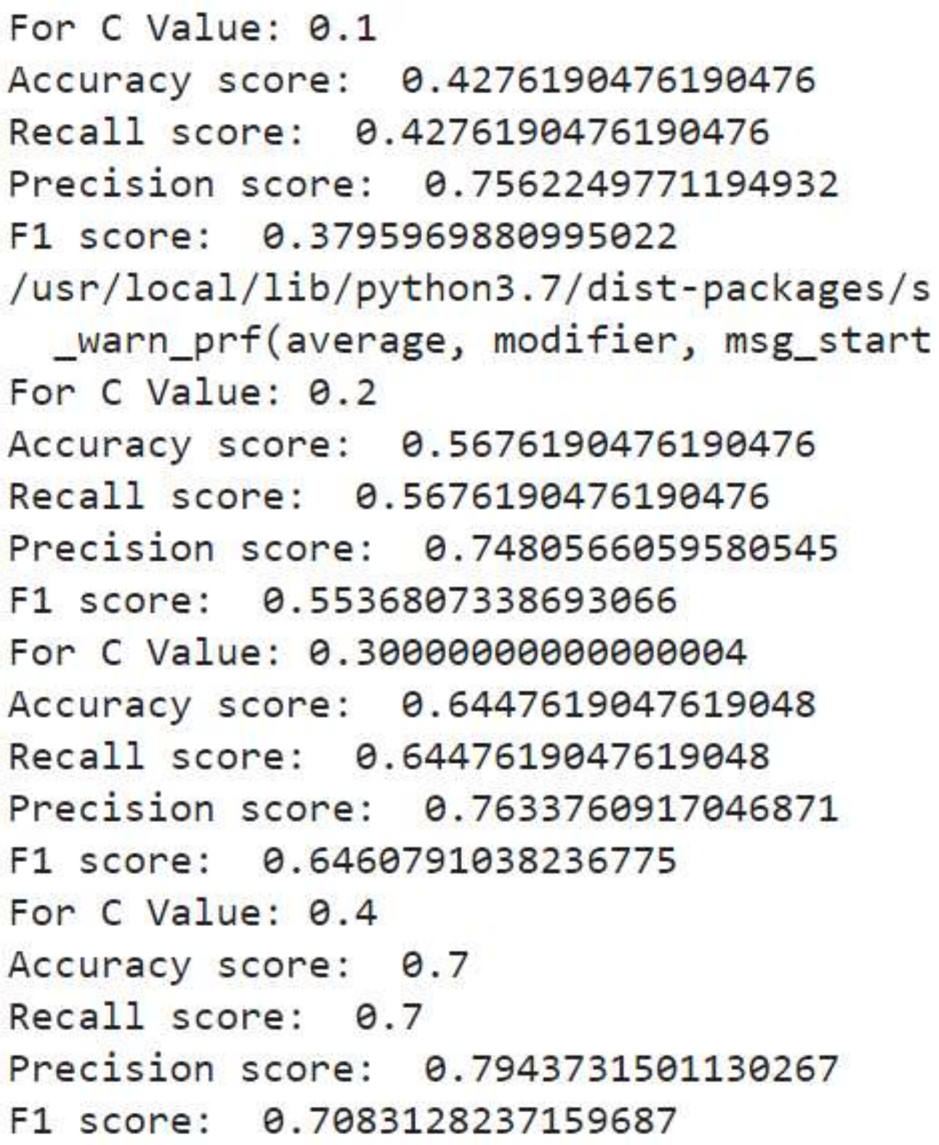
```
Support Vector Machine: SVC
Accuracy score:  0.7952380952380952
Recall score:  0.7952380952380952
Precision score:  0.8311904454423799
F1 score:  0.8016864903375958
```



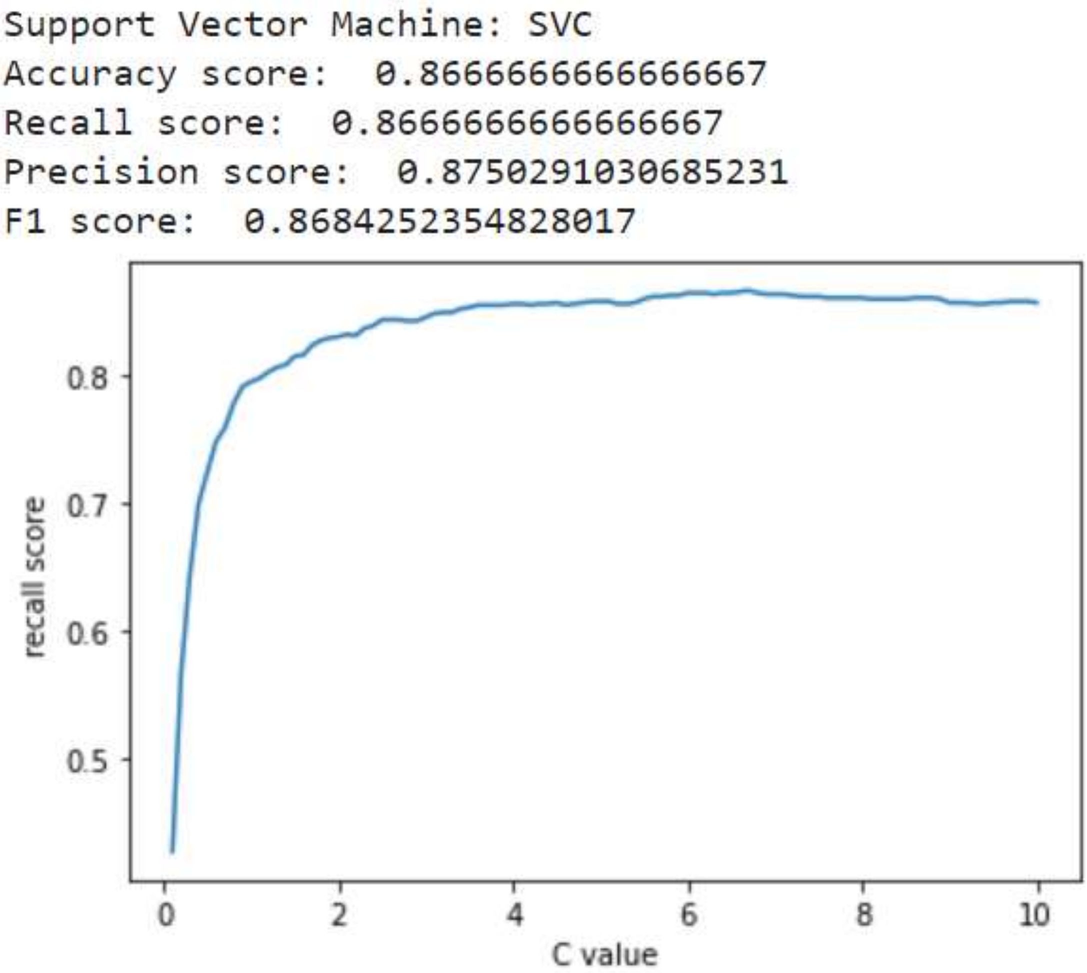
```
# try to adjust c value
C_start = 0.1
C_end = 10
C_inc = 0.1

C_values, recall_scores = [], []

C_val = C_start
best_recall_score = 0
while(C_val < C_end):
    C_values.append(C_val)
    svm_model_svc_loop = svm.SVC(C=C_val, random_state = 42)
    svm_model_svc_loop.fit(training_data, y_train)
    svm_svc_predict_loop_test = svm_model_svc_loop.predict(testing_data)
#     print(lr_predict_loop_test)
    r_score = recall_score(y_test, svm_svc_predict_loop_test, average = 'weighted')
    recall_scores.append(r_score)
    if (r_score > best_recall_score):
        best_recall_score = r_score
        best_lr_predict_test = svm_svc_predict_loop_test
        print("For C Value: " + str(C_val))
        print("Accuracy score: ", accuracy_score(y_test, svm_svc_predict_loop_test))
        print("Recall score: ", recall_score(y_test, svm_svc_predict_loop_test, average = 'weighted'))
        print("Precision score: ", precision_score(y_test, svm_svc_predict_loop_test, average = 'weighted'))
        print("F1 score: ", f1_score(y_test, svm_svc_predict_loop_test, average = 'weighted'))
    C_val = C_val + C_inc
```



```
best_score_C_val = C_values[recall_scores.index(best_recall_score)]
print("1st max value of {0:.3f} occured at C={1:.3f}".format(best_recall_score, best_score_C_val))

plt.plot(C_values, recall_scores, "-")
plt.xlabel("C value")
plt.ylabel("recall score")

svm_model_svc = svm.SVC(C=best_score_C_val, random_state = 42)
svm_model_svc.fit(training_data, y_train)

svm_svc_predictions = svm_model_svc.predict(testing_data)
print("Support Vector Machine: SVC")
print("Accuracy score: ", accuracy_score(y_test, svm_svc_predictions))
print("Recall score: ", recall_score(y_test, svm_svc_predictions, average = 'weighted'))
print("Precision score: ", precision_score(y_test, svm_svc_predictions, average = 'weighted'))
print("F1 score: ", f1_score(y_test, svm_svc_predictions, average = 'weighted'))
```



```
For C Value: 0.1
Accuracy score:  0.4276190476190476
Recall score:  0.4276190476190476
Precision score:  0.7562249771194932
F1 score:  0.3795969880995022
/usr/local/lib/python3.7/dist-packages/s
  _warn_prf(average, modifier, msg_start
For C Value: 0.2
Accuracy score:  0.5676190476190476
Recall score:  0.5676190476190476
Precision score:  0.7480566059580545
F1 score:  0.5536807338693066
For C Value: 0.30000000000000004
Accuracy score:  0.6447619047619048
Recall score:  0.6447619047619048
Precision score:  0.7633760917046871
F1 score:  0.6460791038236775
For C Value: 0.4
Accuracy score:  0.7
Recall score:  0.7
Precision score:  0.7943731501130267
F1 score:  0.7083128237159687
```



```
Support Vector Machine: SVC
Accuracy score:  0.8666666666666667
Recall score:  0.8666666666666667
Precision score:  0.8750291030685231
F1 score:  0.8684252354828017
```

CHAPTER 7

CONCLUSION AND FUTURE WORKS

7.1 Conclusion

With increased number people, enormous crime is reported in the printing and electronics media in a regular interval around the globe and in Bangladesh as well. We can see that sometimes similar crime is occurring repeatedly in same area which can be marked as probable crime zone. So crime information extraction has become a rudimentary task to prevent such crime before occurring. Fortunately, we can find some of the crime news in digital format through our online newspaper which can be utilized for future prediction and protection as well. With this view, this paper aims to extract crime information from online Bangla newspaper. To achieve the task, we developed a Bangla crime corpus.

7.2 Future Work

In future we will normalize the dataset and enrich our newly developed corpus with enormous data to achieve desire accuracy. Our main interested is to extract specific crime information like identify the criminal, victim, crime zone, crime frequency etc. from online Bangla news articles. For this purpose, currently we are annotating our newly developed dataset of 0.1 million words to trained our own Name Entity Recognition (NER) model. Hopefully this work will lead us to reach our intended goal.

REFERENCES

[1] Salma Tabashum, “Performance Analysis of Most Prominent Machine Learning and Deep Learning Algorithms In Classifying Bangla Crime News Articles”, 05-07 June 2020, [Online]. Available: Performance Analysis of Most Prominent Machine Learning and Deep Learning Algorithms In Classifying Bangla Crime News Articles

[2] Mandal A, Sen R. Supervised Learning Methods for Bangla Web Document Categorization. International Journal of Artificial Intelligence & Applications. 2014

[3] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS 2013.

[4] Mandal A, Sen R. Supervised Learning Methods for Bangla Web Document Categorization. International Journal of Artificial Intelligence & Applications. 2014;5(5):93-105.

[5] J. S. D. Bruin, T. K. Cocx, W. A. Kosters, J. F. J. Laros, and J. N.Kok, “Data mining approaches to criminal career analysis,” in SixthInternational Conference on Data Mining (ICDM’06), pp. 171–177, Dec2006.

[6] Kim S. B. , Rim H. C. , Yook D. S. and Lim H. S. , "Effective Methods for Improving Naive Bayes Text Classifiers", LNAI 2417, 2002, pp. 414-423.

[7] F. Sebastiani, "Machine learning in automated text categorization,"2002.

[8] C. Cortes and V. Vapnik, "Support-vector networks," Machine learning

[9] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," Computational linguistics, vol. 18, pp. 467-479, 1992.