

Analysis of COVID-19 Rebound Based on Natural Language

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

by

Sree Vaishnava Reddy Y.V (40310006)



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

SATHYABAMA

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

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI-600119

MAY - 2022

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of
Sree Vaishnava Reddy Y V (40310006) who carried out the project
entitled “Analysis of COVID-19 Rebound Based on Natural
Language Processing” under my / our supervision from November-
2021 to may-2022

Internal Guide

Dr.A.YOVAN FELIX M.E.,Ph.D.,

Head of the Department

Dr.S.Vigneshwari M.E.,Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I Sree Vaishnava Reddy Y.V hereby declare that the Project Report entitled “Analysis of COVID-19 Rebound Based on Natural Language Processing” done by me under the guidance of Dr.A.YOVAN FELIX M.E.,Ph.D. Sathyabama Institute of Science and technology is submitted in partial fulfilment of the requirements for the award of Master of Engineering in Computer Science and Engineering

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to submit my sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

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

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.A.YOVAN FELIX M.E.,Ph.D.**, for his valuable guidance, suggestions, and constant encouragement which paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT OF THE WORK

COVID-19, Corona Virus Disease-2019, caused by a novel Severe Acute Respiratory Syndrome Corona Virus 2 (SARS-CoV-2). Effective screening of this virus can enable quick and efficient diagnosis of COVID-19 can reduce the burden on the healthcare system. Detailed analysis on the provided dataset can build different and various types of machine learning algorithms, which their performance could be computed and further evaluated. In the following case, Random Forest outperformed all the other Machine Learning models like SVR, Xgboost models.

CONTENTS WITH PAGE NUMBERS

TABLE OF CONTENT

CHAPTER NO NO	TITLE	PAGE
I.	INTRODUCTION	1
II.	LITERATURE SURVEY	2
III.	AIM	3
	3.1 OBJECTIVE	3
	3.2 SCOPE OF THE PRESENT INVESTIGATION	3
	3.3 EXISTING SYSTEM	3
	3.4 PROPOSED SYSTEM	4
IV.	EXPERIMENTAL OR MATERIALS AND METHODS	5
	4.1 PYTHON OVERVIEW	5
	4.2 PYTHON ENVIRONMENT	7
	4.3 METHODOLOGY	33
V.	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	45
VI.	SUMMARY AND CONCLUSIONS	47
	REFERENCES	

LIST OF FIGURES:

Figure1: Natural Language Processing (NLP)	15
Figure2: Machine Learning vs. Traditional Programming	19
Figure 3: Learning Phase	20
Figure 4: Inference from Model	20
Figure 5: Machine Learning Algorithms	21
Figure 6: Graph of AL,ML,DL	30
Figure7: System Architecture	33
Figure 8: Data Flow Diagram	38
Figure 9: Use Case Diagram	39
Figure 10: Class Diagram	40
Figure 11: Sequence Diagram	41
Figure 12: Activity Diagram	42
Figure 13: Colloboration Diagram	43

LIST OF TABLES:

Table1: Literature Survey	2
Table 2: Classification	23
Table3: Unsupervised learning	24
Table 4: Difference between Machine Learning and Deep Learning	29
Table 5: ML vs DL	29

CHAPTER 1

INTRODUCTION

The novel coronavirus (COVID-19) hit our blue planet and became an ongoing global pandemic [1]. In a little over six months since the virus was first spotted in mainland China, it has spread to more than 180 countries, infected more than 18.4 million people, and taken more than 692,000 lives as reported in the first week of August 2020. As governments and health organizations scramble to contain the spread of coronavirus, they need all the help they can get, including from artificial intelligence (AI). Though the current AI technologies are far from replicating human intelligence, they are proving to be helpful in tracking the outbreak, diagnosing patients, disinfecting areas, and speeding up the process of finding a cure for COVID-19. Forecasting is a collection of quantitative, probabilistic statements based on historical observation. It is a process of predicting unobserved events and trends. Population health monitoring and forecasting, including epidemiological outbreaks may not have any clinical utility but can be useful tool for planning, decision making and prevention

Type 2 mellitus (DM) is the most common noncommunicable disease (NCD) on the globe; type 2 diabetes affects up to 90% of individuals (T2DM) The risk of T2DM infection is linked to a variety of mental, behavioural, and natural dangers and health issues. Diabetes is a collection of metabolic problems that result in elevated glucose levels over a lengthy amount of time. Diabetes is characterised by a variety of metabolic disorders caused by abnormal insulin release and/or work. Unreasonable cravings, constant thirst, and general desire are among symptoms of aldohexose. If diabetes is not treated, it can lead to serious medical problems including as diabetic, hyperosmolar hyperglycemic corrosive, and even death. Nervousness, brain paralyze, slowness, foot ulcers, and eye anguish might all occur at the same time. Diabetes develops when the body's organs are unable to provide enough insulin, or if the body's basic elements as well as the insulin they create are not utilised. Insulin-incited diabetics is characterised along the a tubular function that manufactures insulin but does not meet the body's ments

(IDDM). Insulin blockage describes type 2 diabetes, which happens when the body's defensive system responds to insulin uniquely in contrast to anticipated. At the point when there isn't sufficient insulin in the body, this can happen. This is known as "non-insulin diabetes mellitus" "adults with diabetes mellitus Diabetics are more likely among those who have a higher Metabolic rate or are overweight. poor academic performance. Physiologically, diabetes has a third standard. In general, the amount of aldohexose in a normal person ranges from 70 to 99 milligrammes per deciliter. Pre-diabetic patients have aldohexosis levels ranging from one to 25 mg/dL. Individuals with high cholesterol but low long-term hypotension of 40 mg/dL were shown to have a variety of ovarian growths.

CHAPTER 2

LITERATURE SURVEY

In this peer-reviewed study, Talha Mahboob Alam and al discuss diabetes in terms of important characteristics, as well as the relationship between different civilizations. To choose practises, assess, predict, and develop affiliation criteria, many methods are used. In the inspection of important components, key items were determined. The findings of the study demonstrate that diabetes is strongly linked to body mass (BMI) and, of course, glucose levels. The downside of studying is that it takes time. information acquired is arranged; however, this method is used in numerous areas of medicine to investigate different types of cancer Parkinson's disease, psoriasis, and.. infection. Diabetes testing should also take into account characteristics such as real occupation, family heritage, and smoking habits [2]. The goal of Deepti Sisodia et al's is to create a version that clearly demonstrated how a sufferer might cultivate diabetes. As a result, three AI computations, such as Decision Graph, SVM, and Naive Bayes, were used in this study to distinguish early diabetes. The review is being undertaken by the UCI Machine Training Foundation's (PIDD). One of the most appealing aspects of using Dataset is that it may be planned. The most important The purpose of this study was to predict diabetic patients who will use the WEKA device and also the PIDD medical library. The limitations of this paradigm include the fact that certain information has not been taken into account while examining the diabetic class [3]. This method amounted to examining at all of the data right away, considerably reducing the number of live humans required to properly assess the patient. As a result, physicians may find it useful to contribute to a better understanding about AD as a people. In any event, therapy at this time will almost certainly be beneficial and advantageous in advancing the evaluation cycle. [4]. The main goal for this work, according to Minzhen Wang et al, is to provide a logical basis to a forecasting of expansion of mouth, hand, and foot disease by focusing on the consequences of environmental changes on the progression of hand, foot, and mouth disease there is a mouth. China. The disease and climatic variables are combined here, as well as The inquiry is ongoing. associated with the line and

then how to modify it [5]. Zhang Y, Long JD, and colleagues proposed the Ghastly data, which is commonly used to assess the severity of various illnesses. In any event, such information has not been analysed in order to classify irresistible diseases. The study's purpose was to come up with a list of illnesses that might identify a class of leaf-borne diseases just at DS level [6]. Another quantifiable strategy has been devised for break its four voice of cardiovascular medicine, which is a means of revealing the health of the heart, as Hu Yuliang et al discuss more easily assess and treat cardiovascular illnesses. The study's subject is a common cardiac problem with symptoms as infection as a result of which it may be difficult to properly calculate and implement technique in suitable locations. [7]. In this research, Chengyu Liu¹ et al discuss the importance of services and amenities portrayals on non-direct percentages of cardio vascular alterations in a group of patients with risk factors. Demonstrating speed, detection wavering inspection, and similar entropy were the three strategies used. Long-term episodes linked to pulse were measured using DFA. We will use several ways for acquiring diabetic data to discriminate passings and passings, according to G. Krishnaveni, Prof. T.Sudha/ 2018 [8]. Diabetes mellitus is designed to make the most prevalent method of getting rid of it easier. The KNN computation, Nave Bayes, and also the vector machine are used to predict using the pre-handled data. These classes have a lot of potential in bioinformatics. RBF, KNN, Nave Bayes, Support Vector machine (svm, and more algorithms are dissected and explained in detail alongside the atomic line. Talha Mahboob Alam, Muhammad Atiflqbal, Yasir Ali, and Abdul Wahab[9] are some of the most well-known people in Pakistan. Here In addition, a number of attributes are described. To choose important behaviors, diabetes, predict, and establish affiliation rules, several gadgets are used. In the research of vital sections, the key items were chosen. The study's findings demonstrate that diabetes is strongly linked to body mass index (BMI) and blood glucose measured in a deductive manner. The goal of this study, according to DeeptiSisodia,Dilip SinghSisodia [10], is to develop a model with a strong predictive incentive for predicting the possibility of a diabetic patient. As a result, three AI computations,in this review to distinguish early diabetes: Decision Tree, SVM, and Naive Bayes.. This Pima Diabetes Data (PIDD) UCI Machines Training Foundation is the target of the audit.

Table1: LITERATURE SURVE

S.No	AUTHOUR	TITTLE	YEAR	ALGORITHM
1	S. Makridakis, E. Spiliotis, and V. Assimakopoulos	Statistical and machine learning forecasting	2018	future forecasting
2	Cheong, K.H, , B.Z.Y.S	Forecasting Hospital Emergency Department Patient Volume	2019	Internet Search Data
3	Huang C, Wang Y, Li X, Ren L, Zhao J, Hu Y	Clinical features of patients infected with 2019 novel coronavirus	2019	healthcare analytics
4	Li Q, Guan X, Wu P, Wang X, Zhou L, Tong Y	Early transmission dynamics	2020	patient data

CHAPTER 3

AIM:

The healthcare industry is a vast industry that requires real time collection and processing of medical data.

3.1 OBJECTIVE:

This proposed model is also compared with other traditional classification algorithms such as Existing techniques, SVR and Random forest and XGBOOST with respect to accuracy, precision and recall.

3.2 SCOPE OF THE PRESENT INVESTIGATION:

The healthcare industry is a vast industry that requires real time collection and processing of medical data. Moreover, at the core of this industry lies the problem of data handling which requires real time prediction and dissemination of information to practitioners for quick medical attention. Major actors of this industry, such as physicians, vendors, hospitals, and health-based companies have attempted to collect, manage, and revive data with the aim of using it to enhance medical practices and for technological innovation. However, dealing with healthcare data has, of late, become a complex task due to the massive volume of the data, security issues, wireless network application incompetence, and the velocity at which it is increasing. Thus, to increase the efficiency, accuracy, and workflow healthcare industries need data analytics tools to manage such complex data

3.3 Existing System:

Typically, the existing mechanisms assumed that the accuracy of prediction was achieved. But this wasn't the case then, hence, it must be improved further to increase the classification accuracy. Also, other research works addressed these issues by introducing efficient combination. Existing Models based on feature selection and classification raised some issues regarding with training dataset and Test dataset.

Dis-Advantage:

- Certain approaches being applicable only for small data.

- Certain combination of classifier over fit with data set while others are under fit.
- Some approaches are not adoptable for real time collection of database implementation.

3.3 Proposed System:

This model is proposed with an algorithm to predict the same. Data pertaining to the study were collected from Kaggle for which the covid-19 analysis and prediction is done, and also suitable data pre-processing methods were applied. This proposed model is also compared with other traditional classification algorithms such as Existing techniques, SVR and Random forest and XGBOOST with respect to accuracy, precision and recall.

Advantage:

1. Improve the accuracy score
2. Deal with large amount dataset

Hardware

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

Software

- Python
- Anaconda

CHAPTER 4

EXPERIMENTAL OR MATERIALS AND METHODS; ALGORITHMS USED

4.1 PYTHON OVERVIEW

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

Python is Interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

Python is Interactive: You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

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

Python is a Beginner's Language: Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, Unix shell, and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include:

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

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

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

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

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

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

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

Databases: Python provides interfaces to all major commercial databases.

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

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

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

IT supports functional and structured programming methods as well as OOP.

It can be used as a scripting language or can be compiled to byte-code for building large applications.

It provides very high-level dynamic data types and supports dynamic type checking.

IT supports automatic garbage collection.

It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

4.2 PYTHON ENVIRONMENT

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

ANACONDA NAVIGATOR

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

Why use Navigator?

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

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

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

WHAT APPLICATIONS CAN I ACCESS USING NAVIGATOR?

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QTConsole
- Spyder
- VSCode
- Glueviz
- Orange 3 App
- Rodeo
- RStudio

Advanced conda users can also build your own Navigator applications

How can I run code with Navigator?

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

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

What's new in 1.9?

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

PYTHON

Python

Python is a general-purpose, versatile and popular programming language. It's great as a first language because it is concise and easy to read, and it is also a good language to have in any programmer's stack as it can be used for everything from web development to software development and scientific applications.

It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time.

Features of Python:

A simple language which is easier to learn, Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax. If you are a newbie, it's a great choice to start your journey with Python.

- **Free and open source**

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software's written in it, you can even make changes to the Python's source code. Python has a large community constantly improving it in each iteration.

Portability

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

Extensible and Embeddable

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code. This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

A high-level, interpreted language

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower level operations.

Large standard libraries to solve common tasks

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server you can use MySQLdb library using import MySQL db Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

Object-oriented

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating object

Python

History and Versions:

Python is predominantly a dynamic typed programming language which was initiated by Guido van Rossum in the year 1989. The major design philosophy that was given more importance was the readability of the code and expressing an idea in fewer lines of code rather than the verbose way of expressing things as in C++ and Java [K-8][K-9]. The other design philosophy that was worth mentioning was that, there should be always a single way and a single obvious way to express a given task which is contradictory to other languages such as C++, Perl etc. [K-10]. Python compiles to an intermediary code and this in turn is interpreted by the Python Runtime Environment to the Native Machine Code. The initial versions of Python were heavily inspired from lisp (for functional programming constructs). Python had heavily borrowed the module system, exception model and also keyword arguments from Modula-3 language [K-10]. Python's developers strive not to entertain premature optimization, even though it might increase the

performance by a few basis points [K-9]. During its design, the creators had conceptualized the language as being a very extensible language, and hence they had designed the language to have a small core library which was extended by a huge standard library [K-7]. Thus as a result, python is used as a scripting language as it can be easily embedded into any application, though it can be used to develop a full-fledged application. The reference implementation of python is CPython. There are also other implementations like Jython, Iron Python which can use python syntax as well as can use any class of Java (Jython) or .Net class (Iron Python). Versions: Python has two versions 2.x version and 3.x version. The 3.x version is a backward incompatible release was released to fix many design issues which plagued the 2.x series. The latest in the 2.x series is 2.7.6 and the latest in 3.x series is 3.4.0. 1.5.2 Paradigms: Python supports multi-paradigms such as: Object-Oriented, Imperative, Functional, Procedural, and Reflective. In Object-Oriented Paradigm, Python supports most of the OOPs concepts such as Inheritance (It also has support for Multiple Inheritance), Polymorphism but its lack of support for encapsulation is a blatant omission as Python doesn't have private, protected members: all class members are public [K-11]. Earlier Python 2.6 versions didn't support some OOP's concepts such as Abstraction through Interfaces and Abstract Classes [K-19]. It also supports Concurrent paradigm, but with Python we will not be able to make truly multitasking applications as the inbuilt threading API is limited by GIL (Global Interpreter Lock) and hence applications that use the threading API cannot run on multi-core parallelly [K-12]. The only remedy is that, the user has to either use the multi-processing module which would fork processes or use Interpreters that haven't implemented GIL such as Jython or Iron Python [K-12]. 1.5.3 Compilation, Execution and Memory Management: Compilation, Execution and Memory Management: 21 A Comparative Studies of Programming Languages (Comparative Studies of Six Programming Language) Just like the other Managed Languages, Python compiles to an intermediary code and this in turn is interpreted by the Python Runtime Environment to the Native Machine Code. The reference implementation (i.e. CPython) doesn't come with a JIT compiler because of which the execution speed is slow compared to native programming languages [K-17]. We can use PyPy interpreter as it includes a JIT compiler rather than using the Python interpreter that comes by default with the python language, if speed of execution is

one of the important factors [K-18]. The Python Runtime Environment also takes care of all the allocation and deallocation of memory through the Garbage Collector. When a new object is created, the GC allocates the necessary memory, and once the object goes out of its scope, the GC doesn't release memory immediately but instead it becomes eligible for Garbage Collection, which would eventually release the memory. Typing Strategies: Python is a strongly dynamic typed language. Python 3 also supports optional static typing [K-20]. There are a few advantages in using a dynamic typed language, the most prominent one would be that the code is more readable as there is less code (in other words has less boiler-plate code). But the main disadvantage in having python as a dynamic programming language is that there would be no way to guarantee that a particular piece of code would run successfully for all the different data-types scenarios simply because it had run successfully with one type. Basically, we don't have any means to find out an error in the code, till the code has started running.

1.5.4 Strengths and Weaknesses and Application Areas:

Python is predominantly used as a scripting language used in developing standalone applications that are being developed with Static-Typed languages, because of the flexibility it provides due to its dynamic typed nature. Python favours rapid application development, which qualifies it to be used for prototyping. To a certain extent, Python is also used in developing websites. Due to its dynamic typing and of the presence of a Virtual Machine, there is a considerable overhead which translates to way less performance when we compare with native programming languages [K-13]. And hence it is not suited

NUMPY

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. At the core of the NumPy package, is the ndarray object. This encapsulates n-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays

and the standard Python sequences:

- NumPy arrays have a fixed size at creation, unlike Python lists (which can grow dynamically). Changing the size of an ND array will create a new array and delete the original.
- The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory. The exception: one can have arrays of (Python, including NumPy) objects, thereby allowing for arrays of different sized elements.
- NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences.
- A growing plethora of scientific and mathematical Python-based packages are using NumPy arrays; though these typically support Python-sequence input, they convert such input to NumPy arrays prior to processing, and they often output NumPy arrays. In other words, in order to efficiently use much (perhaps even most) of today's scientific/mathematical Python-based software, just knowing how to use Python's built-in sequence types is insufficient - one also needs to know how to use NumPy arrays. The points about sequence size and speed are particularly important in scientific computing. As a simple example, consider the case of multiplying each element in a 1-D sequence with the corresponding element in another sequence of the same length. If the data are stored in two Python lists, *a* and *b*, we could iterate over each element:

The Numeric Python extensions (NumPy henceforth) is a set of extensions to the Python programming language which allows Python programmers to efficiently manipulate large sets of objects organized in grid-like fashion. These sets of objects are called arrays, and they can have any number of dimensions: one dimensional arrays are similar to standard Python sequences, two-dimensional arrays are similar to matrices from linear algebra. Note that one-dimensional arrays are also different from any other Python sequence, and that two-dimensional matrices are also different from the matrices of linear algebra, in ways which we will mention later in this text. Why are these extensions needed? The core reason is a very prosaic one, and that is that manipulating a set of a million numbers in Python with the standard data structures such as lists, tuples or classes is much too slow and uses too much space. Anything which we can do in NumPy we can do in standard Python – we just may not be alive to see the

program finish. A more subtle reason for these extensions however is that the kinds of operations that programmers typically want to do on arrays, while sometimes very complex, can often be decomposed into a set of fairly standard operations. This decomposition has been developed similarly in many array languages. In some ways, NumPy is simply the application of this experience to the Python language – thus many of the operations described in NumPy work the way they do because experience has shown that way to be a good one, in a variety of contexts. The languages which were used to guide the development of NumPy include the infamous APL family of languages, Basis, MATLAB, FORTRAN, S and S+, and others. This heritage will be obvious to users of NumPy who already have experience with these other languages. This tutorial, however, does not assume any such background, and all that is expected of the reader is a reasonable working knowledge of the standard Python language. This document is the “official” documentation for NumPy. It is both a tutorial and the most authoritative source of information about NumPy with the exception of the source code. The tutorial material will walk you through a set of manipulations of simple, small, arrays of numbers, as well as image files. This choice was made because:

- A concrete data set makes explaining the behavior of some functions much easier to motivate than simply talking about abstract operations on abstract data sets;
- Every reader will at least have an intuition as to the meaning of the data and organization of image files, and
- The result of various manipulations can be displayed simply since the data set has a natural graphical representation.

All users of NumPy, whether interested in image processing or not, are encouraged to follow the tutorial with a working NumPy installation at their side, testing the examples, and, more importantly, transferring the understanding gained by working on images to their specific domain. The best way to learn is by doing – the aim of this tutorial is to guide you along this “doing.”

Domain Overview:

DOMAIN SPECIFICATION:-

NLP

In the early 1900s, a Swiss linguistics professor named Ferdinand de Saussure died, and in the process, almost deprived the world of the concept of “Language as a Science.”

From 1906 to 1911, Professor Saussure offered three courses at the University of Geneva, where he developed an approach describing languages as “systems.” Within the language, a sound represents a concept – a concept that shifts meaning as the context changes.



Figure1: Natural Language Processing (NLP)

He argued that meaning is created inside language, in the relations and differences between its parts. Saussure proposed “meaning” is created within a language’s relationships and contrasts.

A shared language system makes communication possible. Saussure viewed society as a system of “shared” social norms that provides conditions for reasonable, “extended” thinking, resulting in decisions and actions by individuals. (The same view can be applied to modern computer languages).

Natural Language Processing (NLP) is an aspect of Artificial Intelligence that helps computers understand, interpret, and utilize human languages. NLP allows computers to communicate with people, using a human language.

Natural Language Processing also provides computers with the ability to read text, hear speech, and interpret it. NLP draws from several disciplines, including computational linguistics and computer science, as it attempts to close the gap between human and computer communications.

Generally speaking, NLP breaks down language into shorter, more basic pieces, called tokens (words, periods, etc.), and attempts to understand the relationships of the tokens. This process often uses higher-level NLP features, such as:

- **Content Categorization:** A linguistic document summary that includes content alerts, duplication detection, search, and indexing.
- **Topic Discovery and Modeling:** Captures the themes and meanings of text collections, and applies advanced analytics to the text.
- **Contextual Extraction:** Automatically pulls structured data from text-based sources.
- **Sentiment Analysis:** Identifies the general mood, or subjective opinions, stored in large amounts of text. Useful for opinion mining.
- **Text-to-Speech and Speech-to-Text Conversion:** Transforms voice commands into text, and vice versa.
- **Document Summarization:** Automatically creates a synopsis, condensing large amounts of text.
- **Machine Translation:** Automatically translates the text or speech of one language into another.

NLP Begins and Stops

Noam Chomsky published his book, *Syntactic Structures*, in 1957. In it, he revolutionized previous linguistic concepts, concluding that for a computer to understand a language, the sentence structure would have to be changed. With this as his goal, Chomsky created a style of grammar called Phase-Structure Grammar, which methodically translated natural language sentences into a format that is usable by computers. (The overall goal was to create a computer capable of imitating the human brain, in terms of in thinking and communicating, or AI.)

In 1958, the programming language LISP (Locator/Identifier Separation Protocol), a computer language still in use today, was released by John McCarthy.

In 1964, ELIZA, a “typewritten” comment and response process, designed to imitate a psychiatrist using reflection techniques, was developed. (It did this by rearranging sentences and following relatively simple grammar rules, but there was no understanding on the computer’s part.) Also in 1964, the U.S. National Research Council (NRC) created the Automatic Language Processing Advisory Committee, or ALPAC, for short. This committee was tasked with evaluating the progress of Natural Language Processing research.

In 1966, the NRC and ALPAC initiated the first AI and NLP stoppage, by halting the funding of research on Natural Language Processing and machine translation. After twelve years of research, and \$20 million dollars, machine translations were still more expensive than manual human translations, and there were still no computers that came anywhere near being able to carry on a basic conversation. In 1966, Artificial Intelligence and Natural Language Processing (NLP) research was considered a dead end by many (though not all).

Return of the NLP

It took nearly fourteen years (until 1980) for Natural Language Processes and Artificial Intelligence research to recover from the broken expectations created by extreme enthusiasts. In some ways, the AI stoppage had initiated a new phase of fresh ideas, with earlier concepts of machine translation being abandoned, and new ideas promoting new research, including expert systems. The mixing of linguistics and statistics, which had been popular in early NLP research, was replaced with a theme of pure statistics. The 1980s initiated a fundamental reorientation, with simple approximations replacing deep analysis, and the evaluation process becoming more rigorous.

Until the 1980s, the majority of NLP systems used complex, “handwritten” rules. But in the late 1980s, a revolution in NLP came about. This was the result of both the steady increase of computational power, and the shift to Machine Learning algorithms. While some of the early Machine Learning algorithms (decision trees provide a good example) produced systems similar to the old school handwritten rules, research has increasingly focused on statistical models. These statistical models are capable making soft, probabilistic decisions. Throughout the 1980s, IBM was responsible for the development of several successful, complicated statistical models.

In the 1990s, the popularity of statistical models for Natural Language Processes analyses rose dramatically. The pure statistics NLP methods have become remarkably valuable in keeping pace with the tremendous flow of online text. N-Grams have become useful, recognizing and tracking clumps of linguistic data, numerically. In 1997, LSTM recurrent neural net (RNN) models were introduced, and found their niche in 2007 for voice and text processing. Currently, neural net models are considered the cutting edge of research and development in the NLP’s understanding of text and speech generation.

MACHINE LEARNING

Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data (i.e., example) to produce accurate results.

Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights. Machine learning is closely related to data mining and Bayesian predictive modeling. The machine receives data as input, use an algorithm to formulate answers.

A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation.

Machine learning is also used for a variety of task like fraud detection, predictive maintenance, portfolio optimization, automatize task and so on.

Machine Learning vs. Traditional Programming

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

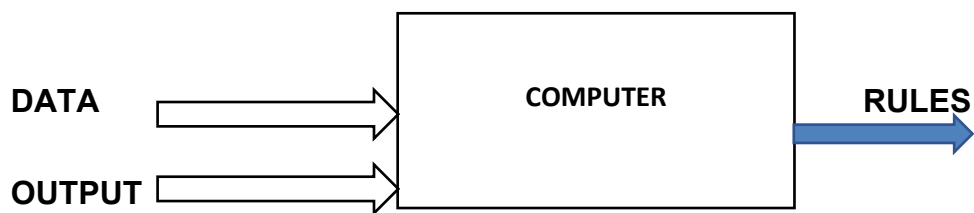


Figure2: Machine Learning vs. Traditional Programming

Machine Learning

How does Machine learning work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feed a previously unseen example, the machine has difficulties to predict.

The core objective of machine learning is the **learning** and **inference**. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the **data**. One crucial part of the data scientist is to choose carefully which data to provide to the machine.

The list of attributes used to solve a problem is called a **feature vector**. You can think of a feature vector as a subset of data that is used to tackle a problem.

The machine uses some fancy algorithms to simplify the reality and transform this discovery into a **model**. Therefore, the learning stage is used to describe the data and summarize it into a model.

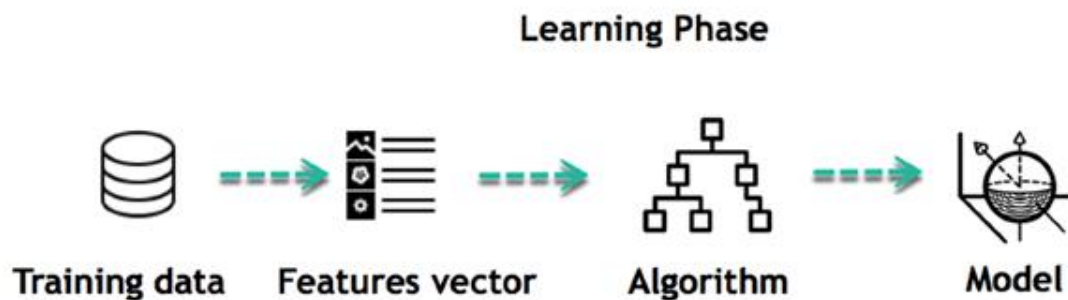


Figure 3: Learning Phase

For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant: This is the model

Inferring

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

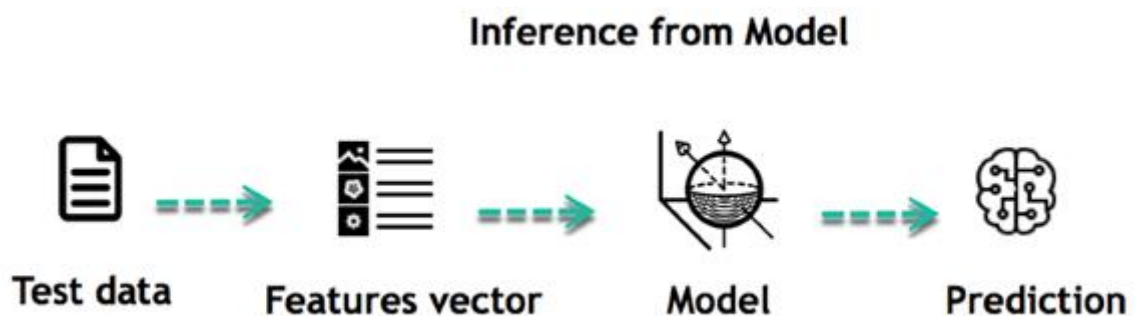


Figure 4: Inference from Model

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

1. Define a question
2. Collect data
3. Visualize data

4. Train algorithm
5. Test the Algorithm
6. Collect feedback
7. Refine the algorithm
8. Loop 4-7 until the results are satisfying
9. Use the model to make a prediction

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

Machine learning Algorithms and where they are used?

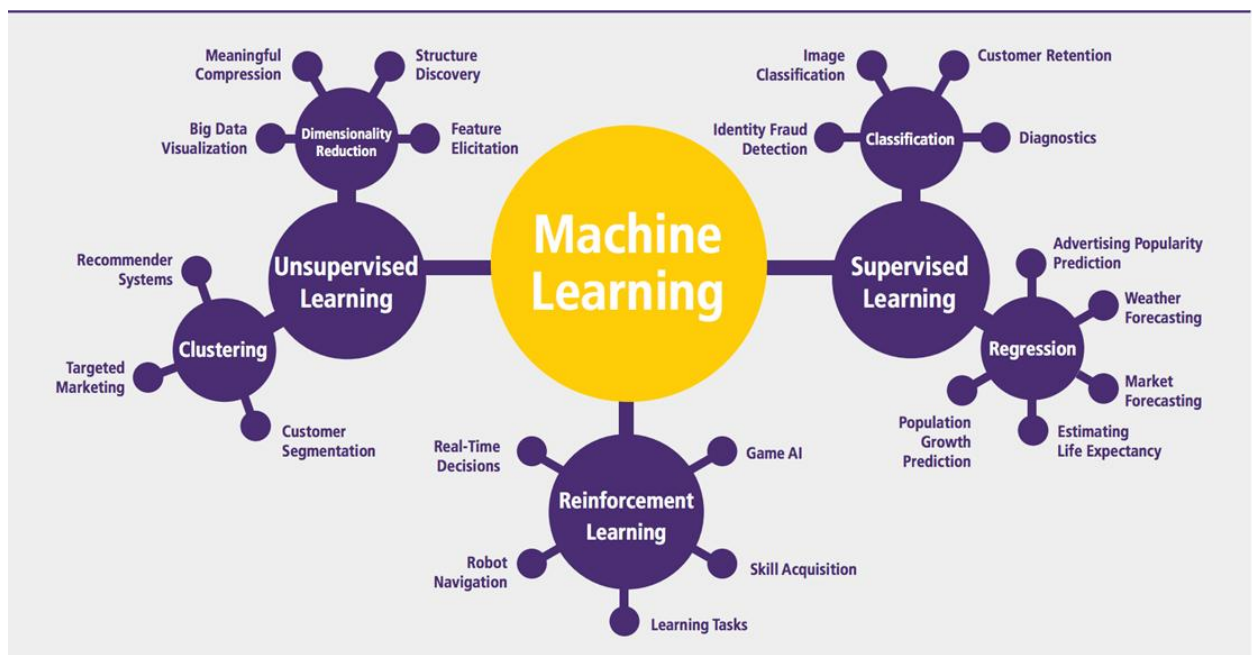


Figure 5: Machine Learning Algorithms

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

Supervised learning

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans. You can use supervised learning when the output data is known. The algorithm will predict new data.

There are two categories of supervised learning:

- Classification task ,Regression task

Classification

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female.

The label can be of two or more classes. The above example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class)

Regression

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

Algorithm	Description	Type
Linear regression	Finds a way to correlate each feature to the output to help predict future values.	Regression
Logistic regression	Extension of linear regression that's used for classification tasks. The output variable is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colors)	Classification

Decision tree	Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes (e.g., if a feature is a color, each possible color becomes a new branch) until a final decision output is made	Regression Classification
Naive Bayes	The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event.	Regression Classification
Support vector machine	Support Vector Machine, or SVM, is typically used for the classification task. SVM algorithm finds a hyperplane that optimally divided the classes. It is best used with a non-linear solver.	Regression (not very common) Classification
Random forest	The algorithm is built upon a decision tree to improve the accuracy drastically. Random forest generates many times simple decision trees and uses the 'majority vote' method to decide on which label to return. For the classification task, the final prediction will be the one with the most votes; while for the regression task, the average prediction of all the trees is the final prediction.	Regression Classification
AdaBoost	Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome	Regression Classification
Gradient-boosting trees	Gradient-boosting trees is a state-of-the-art classification/regression technique. It is focusing on the error committed by the previous trees and tries to correct it.	Regression Classification

Table 2: Classification

Unsupervised learning

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns)

You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you

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

Application of Machine learning

Augmentation:

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

Automation:

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

Finance Industry

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

Government organization

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

Healthcare industry

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

Marketing

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

Example of application of Machine Learning in Supply Chain

Machine learning gives terrific results for visual pattern recognition, opening up many potential applications in physical inspection and maintenance across the entire supply chain network.

Unsupervised learning can quickly search for comparable patterns in the diverse dataset. In turn, the machine can perform quality inspection throughout the logistics hub, shipment with damage and wear.

For instance, IBM's Watson platform can determine shipping container damage. Watson combines visual and systems-based data to track report and make recommendations in real-time.

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

Example of Machine Learning Google Car

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

Deep Learning

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

Reinforcement Learning

Reinforcement learning is a subfield of machine learning in which systems are trained by receiving virtual "rewards" or "punishments," essentially learning by trial and error. Google's DeepMind has used reinforcement learning to beat a human

champion in the Go games. Reinforcement learning is also used in video games to improve the gaming experience by providing smarter bot.

One of the most famous algorithms are:

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

Applications/ Examples of deep learning applications

AI in Finance: The financial technology sector has already started using AI to save time, reduce costs, and add value. Deep learning is changing the lending industry by using more robust credit scoring. Credit decision-makers can use AI for robust credit lending applications to achieve faster, more accurate risk assessment, using machine intelligence to factor in the character and capacity of applicants.

Underwrite is a Fintech company providing an AI solution for credit makers company. underwrite.ai uses AI to detect which applicant is more likely to pay back a loan. Their approach radically outperforms traditional methods.

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

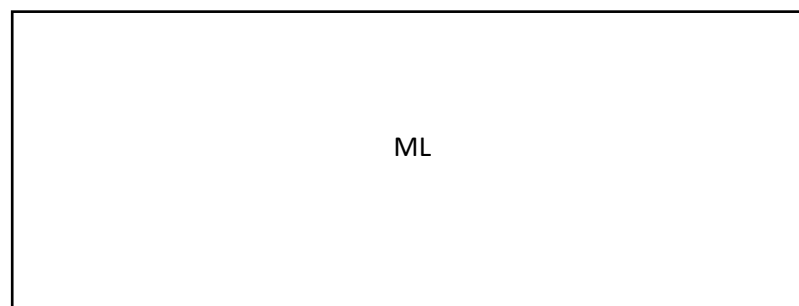
At that time, Under Armour had all of the 'must have' HR technology in place such as transactional solutions for sourcing, applying, tracking and onboarding but those tools weren't useful enough. Under armour choose **HireVue**, an AI provider for HR solution, for both on-demand and live interviews. The results were bluffing; they managed to decrease by 35% the time to fill. In return, the hired higher quality staffs.

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

For example, deep-learning analysis of audio allows systems to assess a customer's emotional tone. If the customer is responding poorly to the AI chatbot, the system can be rerouted the conversation to real, human operators that take over the issue.

Apart from the three examples above, AI is widely used in other sectors/industries.

Artificial Intelligence



Difference between Machine Learning and Deep Learning

	Machine Learning	Deep Learning
Data Dependencies	Excellent performances on a small/medium dataset	Excellent performance on a big dataset
Hardware dependencies	Work on a low-end machine.	Requires powerful machine, preferably with GPU: DL performs a significant amount of matrix multiplication

Feature engineering	Need to understand the features that represent the data	No need to understand the best feature that represents the data
Execution time	From few minutes to hours	Up to weeks. Neural Network needs to compute a significant number of weights
Interpretability	Some algorithms are easy to interpret (logistic, decision tree), some are almost impossible (SVM, XGBoost)	Difficult to impossible

Table 4: Difference between Machine Learning and Deep Learning

When to use ML or DL?

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

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

Table 5: ML vs DL

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

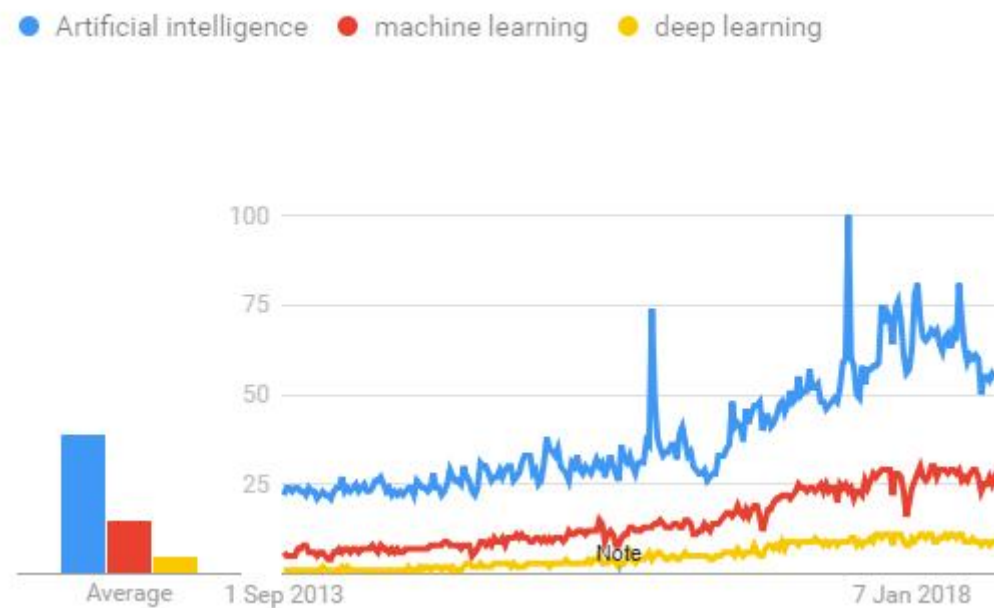


Figure: 6 Graph of AL,ML,DL

TensorFlow

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

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

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

- Researchers
- Data scientists

- Programmers.

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

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

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

In this tutorial, you will learn

TensorFlow Architecture

Tensorflow architecture works in three parts:

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

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

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

Where can Tensorflow run?

TensorFlow can hardware, and software requirements can be classified into Development Phase: This is when you train the mode. Training is usually done on your Desktop or laptop.

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

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

You can train it on multiple machines then you can run it on a different machine, once you have the trained model.

The model can be trained and used on GPUs as well as CPUs. GPUs were initially designed for video games. In late 2010, Stanford researchers found that GPU was also very good at matrix operations and algebra so that it makes them very fast for doing these kinds of calculations. Deep learning relies on a lot of matrix multiplication. TensorFlow is very fast at computing the matrix multiplication because it is written in C++. Although it is implemented in C++, TensorFlow can be accessed and controlled by other languages mainly, Python.

Finally, a significant feature of TensorFlow is the TensorBoard. The TensorBoard enables to monitor graphically and visually what TensorFlow is doing.

List of Prominent Algorithms supported by TensorFlow

- Linear regression: `tf.estimator.LinearRegressor`
- Classification: `tf.estimator.LinearClassifier`
- Deep learning classification: `tf.estimator.DNNClassifier`
- Deep learning wibe and deep: `tf.estimator.DNNLinearCombinedClassifier`
- Booster tree regression: `tf.estimator.BoostedTreesRegressor`
- Boosted tree classification: `tf.estimator.BoostedTreesClassifier`

4.3 METHODOLOGY

IMPLEMENTATION PROCESS:

MODULES

1. DATA COLLECTION
2. DATA PRE-PROCESSING
3. FEATURE EXTRATION
4. EVALUATION MODEL

DATA COLLECTION

Data used in this paper is a set of product reviews collected from credit card transactions records. This step is concerned with selecting the subset of all available data that you will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answer. Data for which you already know the target answer is called *labelled data*.

DATA PRE-PROCESSING

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

Three common data pre-processing steps are:

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

FEATURE EXTRACTION

Next thing is to do Feature extraction is an attribute reduction process. Unlike feature selection, which ranks the existing attributes according to their predictive significance, feature extraction actually transforms the attributes. The transformed attributes, or features, are linear combinations of the original attributes. Finally, our models are trained using Classifier algorithm. We use classify module on Natural Language Toolkit library on Python. We use the

labelled dataset gathered. The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre-processed data. The chosen classifiers were Random forest. These algorithms are very popular in text classification tasks.

EVALUATION MODEL

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

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

System Architecture:

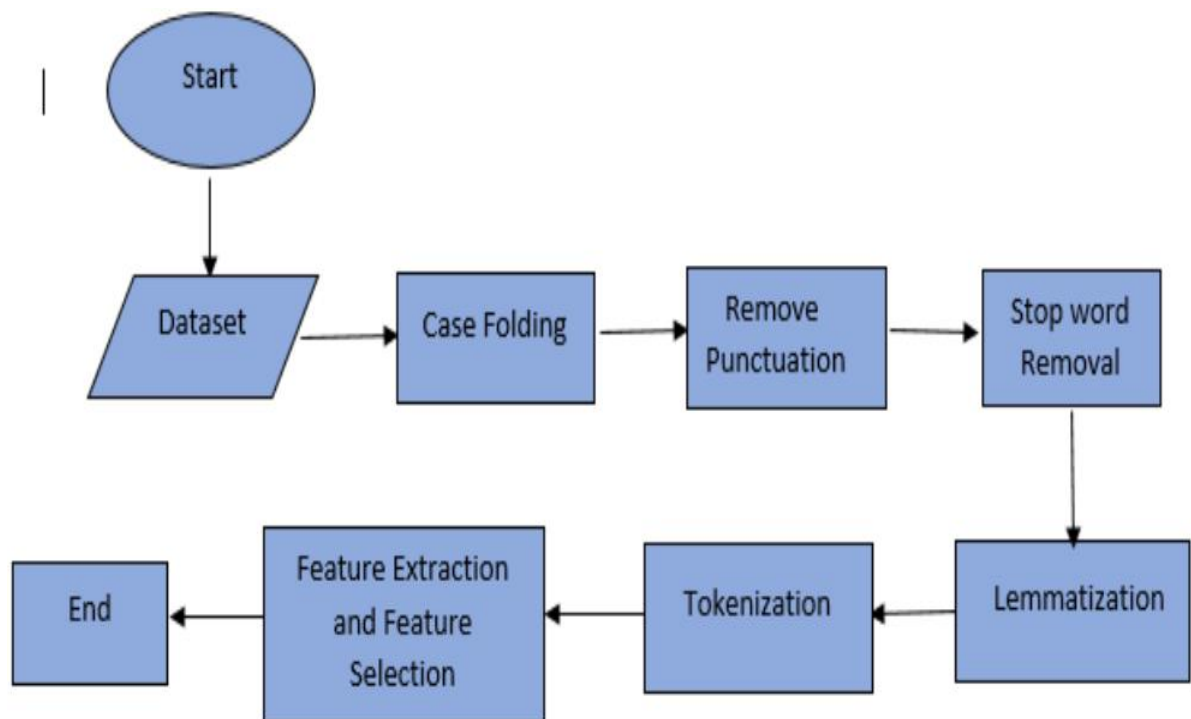
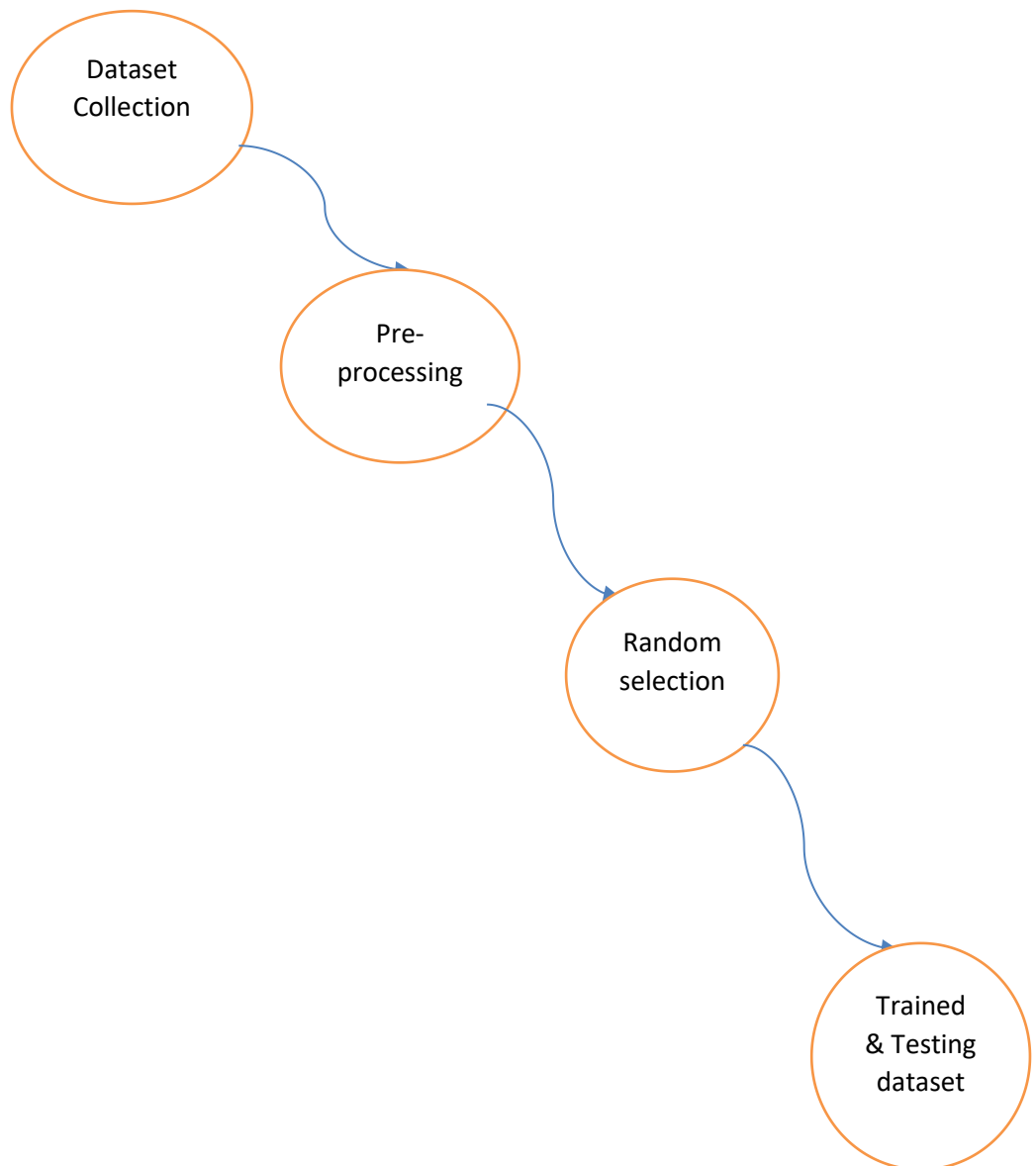


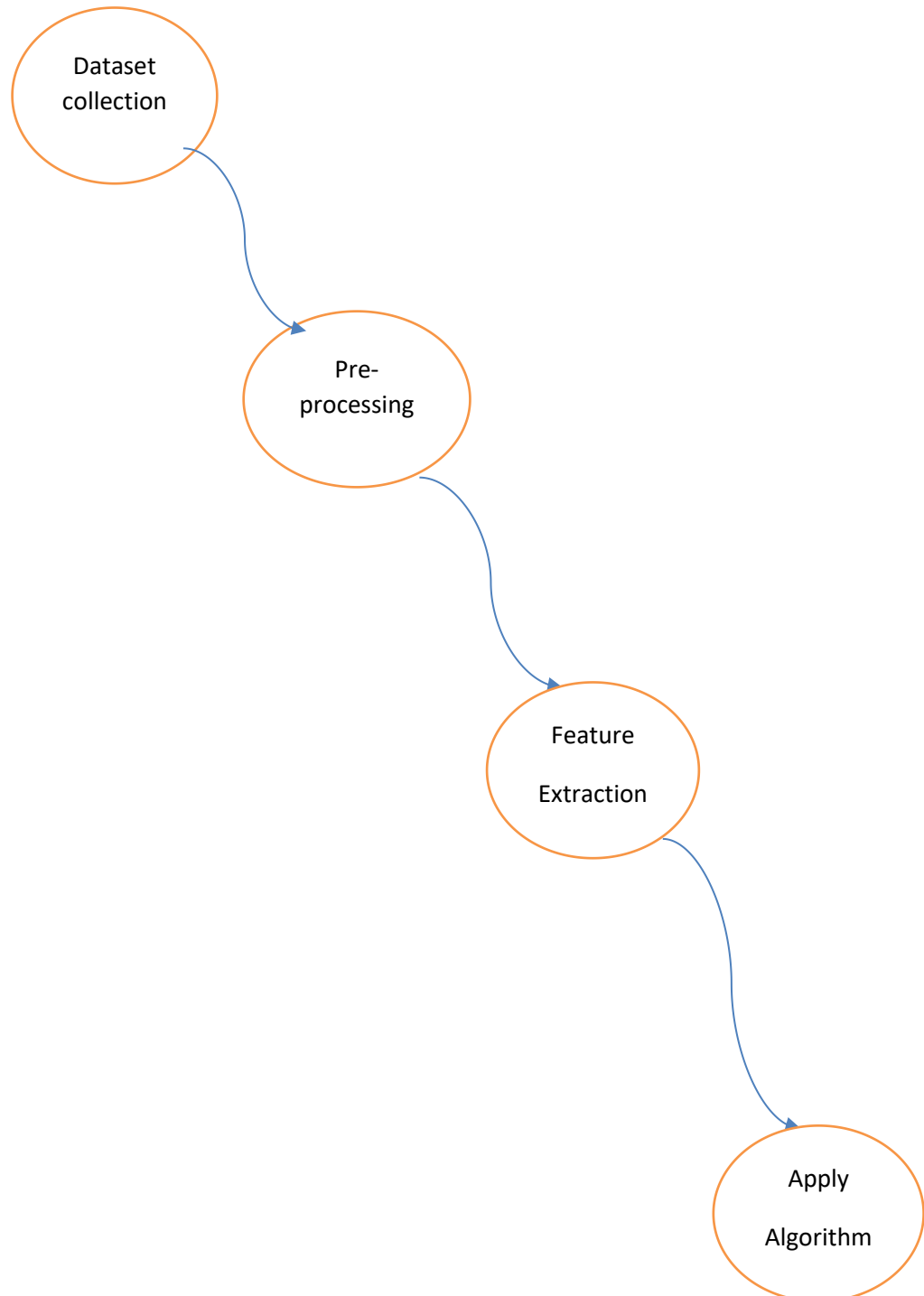
Figure7: System Architecture

DATA FLOW DIAGRAM

LEVEL 0



LEVEL 1



LEVEL 2

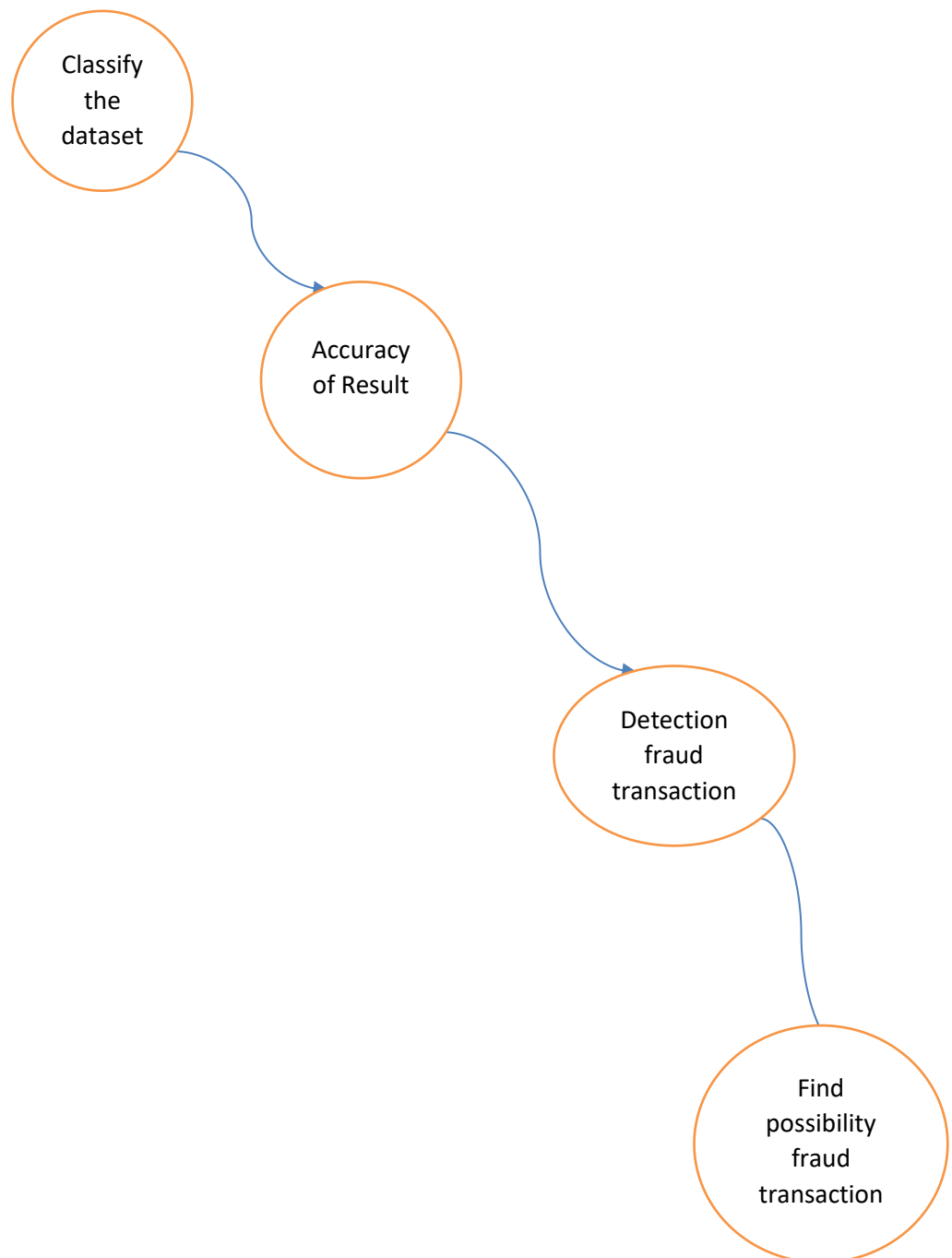


Figure 8: Data Flow Diagram

UML DIAGRAM - USE CASE DIAGRAM:

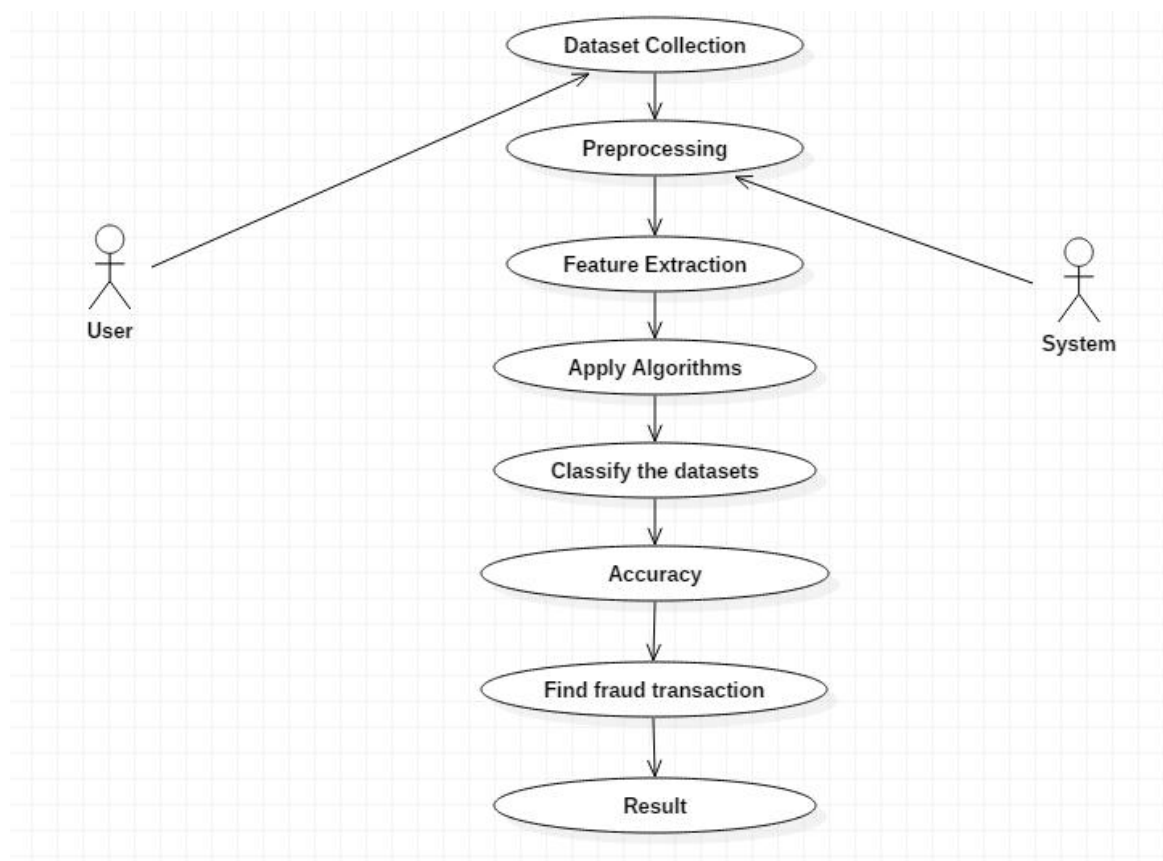


Figure 9: Use Case Diagram

CLASS DIAGRAM:

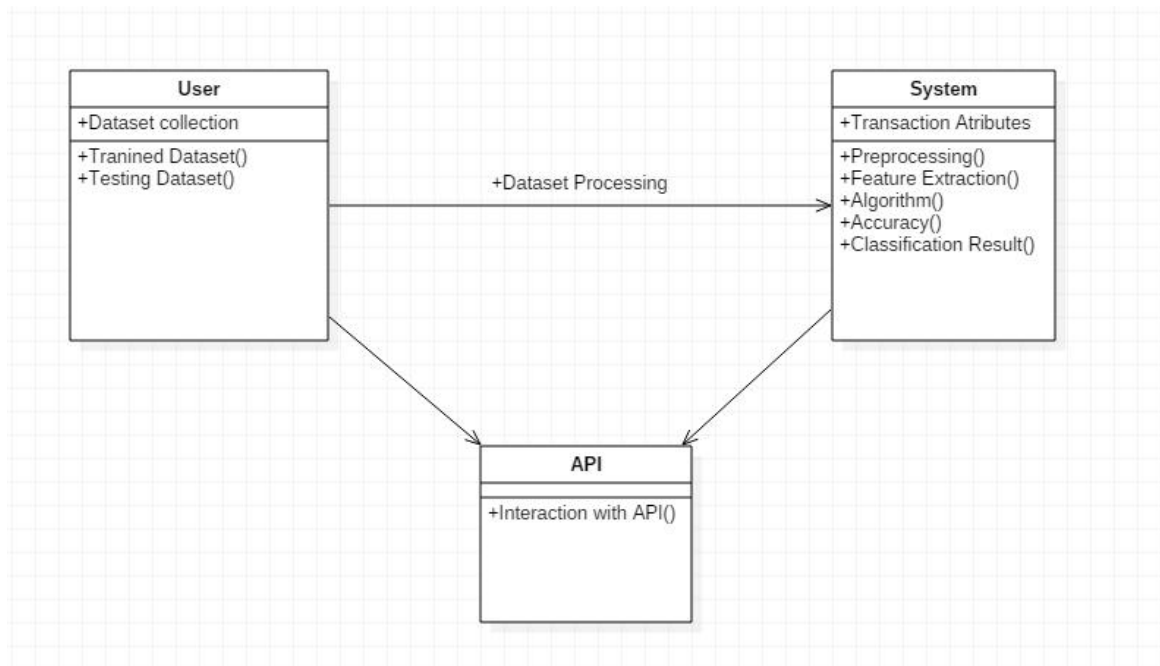


Figure 10: Class Diagram

SEQUENCE DIAGRAM:

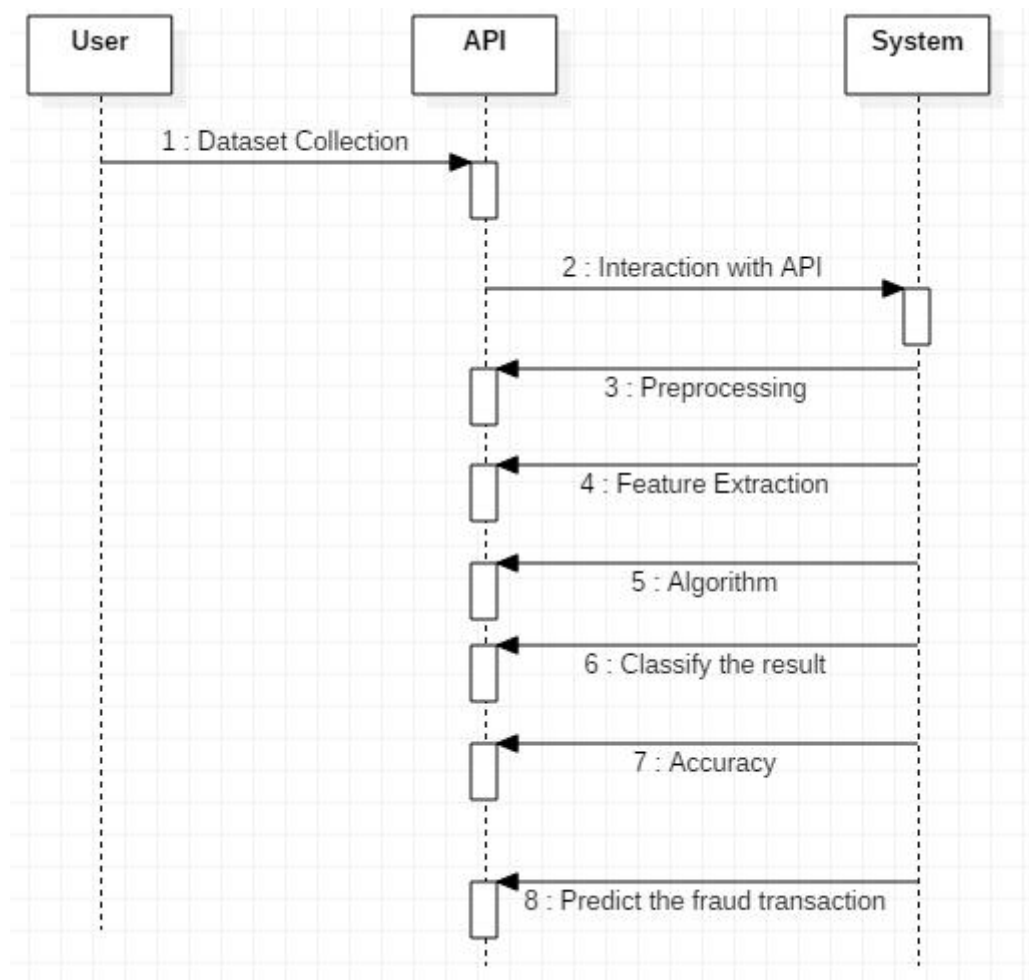


Figure 11: Sequence Diagram

ACTIVITY DIAGRAM:

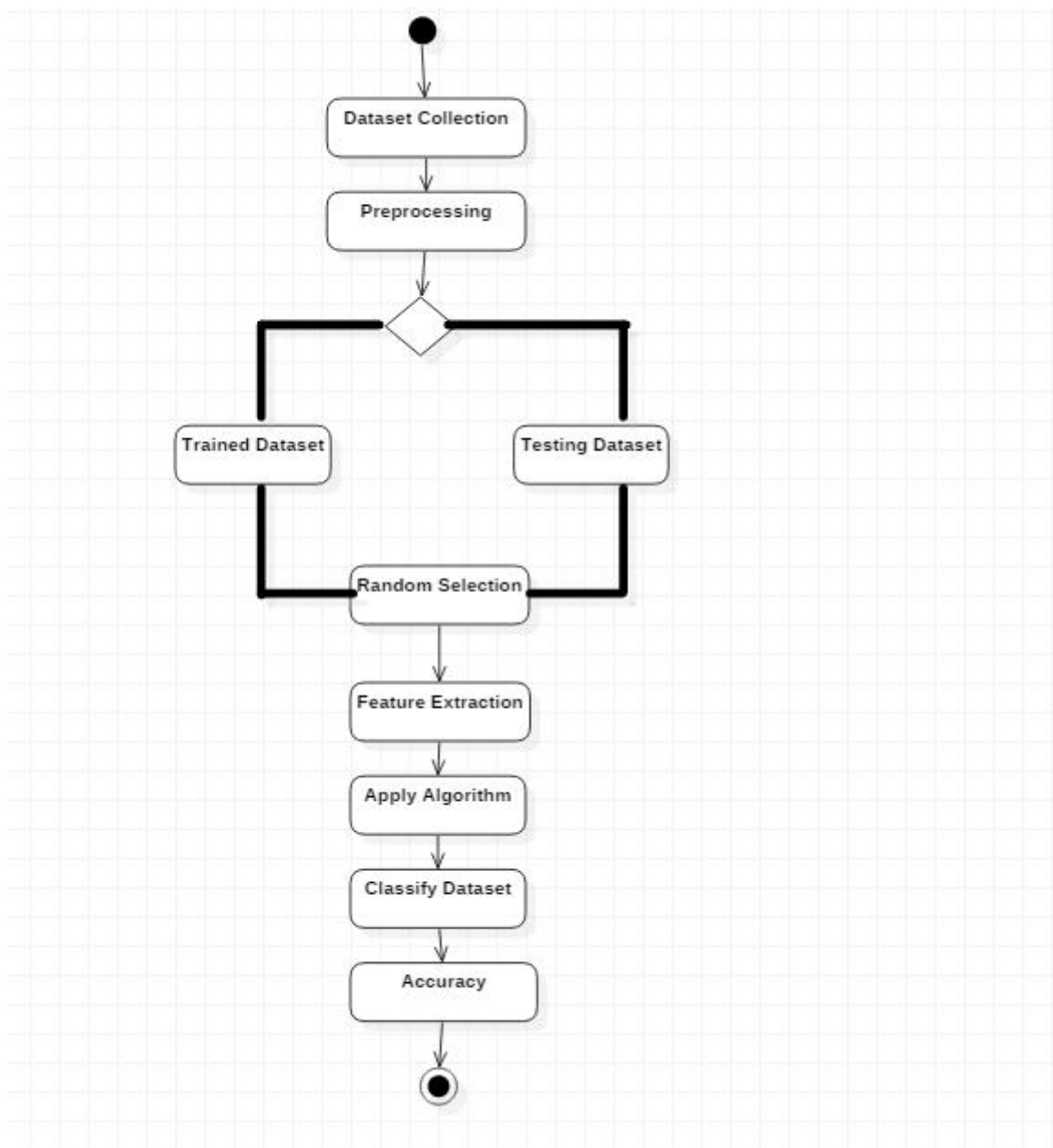


Figure 11: Activity Diagram

COLLOBORATION DIAGRAM:

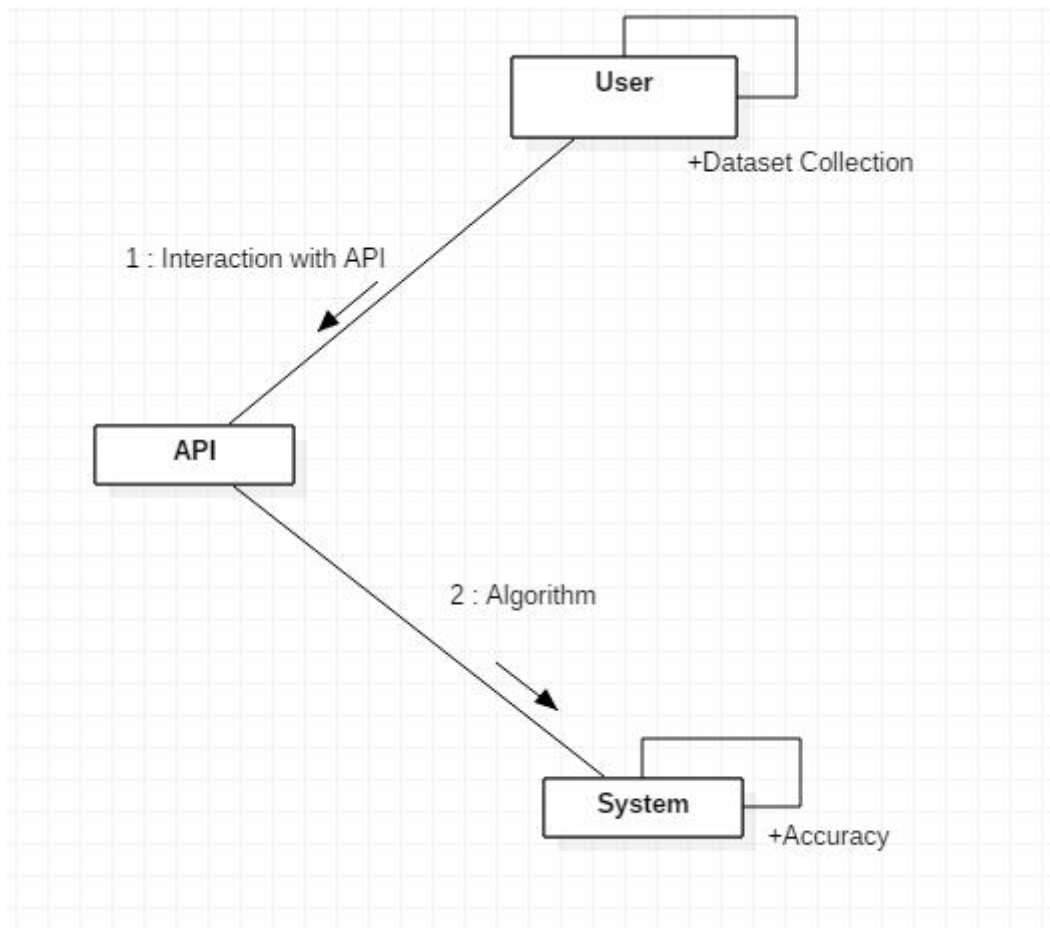


Figure 12: COLLOBORATION DIAGRAM:

ALGORITHMS:

RANDOM FOREST

Random forest is a type of supervised machine learning algorithm based on ensemble learning. Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The random forest algorithm combines multiple algorithm of the same type i.e. multiple decision *trees*, resulting in a *forest of*

trees, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

HOW RANDOM FOREST WORKS

The following are the basic steps involved in performing the random forest algorithm

1. Pick N random records from the dataset.
2. Build a decision tree based on these N records.
3. Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
4. For classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

ADVANTAGES OF USING RANDOM FOREST

Pros of using random forest for classification and regression.

1. The random forest algorithm is not biased, since, there are multiple trees and each tree is trained on a subset of data. Basically, the random forest algorithm relies on the power of "the crowd"; therefore, the overall biasedness of the algorithm is reduced.
2. This algorithm is very stable. Even if a new data point is introduced in the dataset the overall algorithm is not affected much since new data may impact one tree, but it is very hard for it to impact all the trees.
3. The random forest algorithm works well when you have both categorical and numerical features.
4. The random forest algorithm also works well when data has missing values or it has not been scaled we.

CHAPTER 5

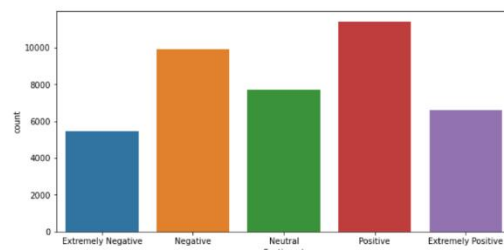
RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

OUTPUTS:

```
In [4]: df.head()
```

Out[4]:

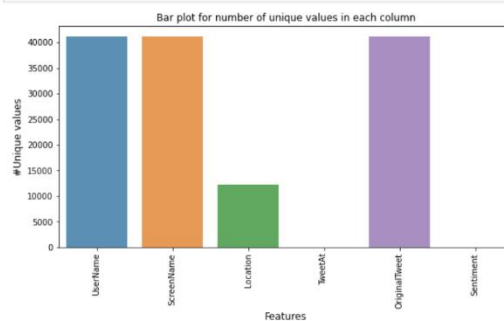
	UserName	ScreenName	Location	TweetAt	OriginalTweet	Sentiment
0	3799	48751	London	16-03-2020	@MeNyrbie @Phil_Gahan @Chrstitv https://t.co/...	Neutral
1	3800	48752	UK	16-03-2020	advice Talk to your neighbours family to excha...	Positive
2	3801	48753	Vagabonds	16-03-2020	Coronavirus Australia: Woolworths to give elde...	Positive
3	3802	48754	NaN	16-03-2020	My food stock is not the only one which is emp...	Positive
4	3803	48755	NaN	16-03-2020	Me, ready to go at supermarket during the #COV...	Extremely Negative



```
In [8]: df.head()
```

Out[8]:

	UserName	ScreenName	Location	TweetAt	OriginalTweet	Sentiment	CleanedTweet
0	3799	48751	London	16-03-2020	@MeNyrbie @Phil_Gahan @Chrstitv https://t.co/...	Neutral	Gahan https t co iFz9FAn2Pa and https ...
1	3800	48752	UK	16-03-2020	advice Talk to your neighbours family to excha...	Positive	advice Talk to your neighbours family to excha...
2	3801	48753	Vagabonds	16-03-2020	Coronavirus Australia: Woolworths to give elde...	Positive	Coronavirus Australia Woolworths to give elde...
3	3802	48754	NaN	16-03-2020	My food stock is not the only one which is emp...	Positive	My food stock is not the only one which is emp...
4	3803	48755	NaN	16-03-2020	Me, ready to go at supermarket during the #COV...	Extremely Negative	Me ready to go at supermarket during the ou...



Activate Windows
Go to Settings to activate Windows.

DataSet:

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

UserNm

ScreenN

Location

TweetAt

OriginalTw

Sentiment

3799

48751

London

16-03-202

@MeNyb

Neutral

3800

48752

UK

16-03-202

advice Tall

Positive

3801

48753

Vagabond

16-03-202

Coronavir

Positive

3802

48754

16-03-202

My food

Positive

3803

48755

16-03-202

Me, ready

Extremely Negative

3804

48756

ÅæT: 36.3

16-03-202

As news of

Positive

3805

48757

35.926541

16-03-202

Cashier at

Positive

3806

48758

Austria

16-03-202

Was at

Neutral

3807

48759

Atlanta, G

16-03-202

Due to CO

Positive

3808

48760

BHAVNAG

16-03-202

For coron

Negative

3809

48761

Makati, M

16-03-202

All month

Neutral

3810

48762

Pitt Mead

16-03-202

Due to

Extremely Positive

3811

48763

Hornings

16-03-202

#hornings

Extremely Positive

3812

48764

Chicago, IL

16-03-202

Me: I

Positive

3813

48765

16-03-202

ADARA Rel

Positive

3814

48766

Houston, T

16-03-202

Lines at

Positive

3815

48767

Saudi Arab

16-03-202

?????

Neutral

3816

48768

Ontario, C

16-03-202

@eyeont

Neutral

3817

48769

North Ame

16-03-202

Amazon

Extremely Positive

3818

48770

Denver, CC

16-03-202

For those

Positive

3819

48771

southamp

16-03-202

with 100

Extremely Negative

3820

48772

Global

16-03-202

https://t

Neutral

3821

48773

16-03-202

We have

Extremely Positive

3822

48774

16-03-202

We have

Extremely Positive

3823

48775

Downstag

16-03-202

@10Down

Negative

3824

48776

London

16-03-202

UK #consu

Extremely Positive

3825

48777

Ketchum, I

16-03-202

In prepara

Negative

3826

48778

Everywher

16-03-202

This morni

Extremely Negative

Corona_NLP_train

Ready

Accessibility: Unavailable

Type here to search

</

			UserName	ScreenN	Nar Location	TweetAt	OriginalTw	Sentiment	CleanedTw	label	tfidf	kmeans																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																						
--	--	--	----------	---------	--------------	---------	------------	-----------	-----------	-------	-------	--------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

CHAPTER 6

CONCLUSIONS

The application of Artificial Intelligence is very crucial to process patient data for efficient treatment strategies. In this paper we presented a model that implements the Random Forest algorithm boosted by the AdaBoost algorithm, with a F1 Score of 0.86 on the COVID-19 patient dataset. We have discovered that the Boosted Random Forest algorithm provides accurate predictions even on imbalanced datasets. The data analyzed in this study has revealed that death rates were higher amongst the Wuhan natives compared to non-natives. Also, male patients had a greater death rate compared to female patients. The majority of affected patients are aged between of 20 and 70 years.

REFERENCES

1. Huang C, Wang Y, Li X, Ren L, Zhao J, Hu Y, et al. Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *Lancet*. (2020) 395:497–506. doi: 10.1016/S0140-6736(20)30183-5
2. Li Q, Guan X, Wu P, Wang X, Zhou L, Tong Y, et al. Early transmission dynamics in Wuhan, China, of novel coronavirus–infected pneumonia. *N Engl J Med*. (2020) 382:1199–207. doi: 10.1056/NEJMoa2001316
3. Chen N, Zhou M, Dong X, Qu J, Gong F, Han Y, et al. Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: a descriptive study. *Lancet*. (2020) 395:507–13. doi: 10.1016/S0140-6736(20)30211-7
4. Clinical Management of Severe Acute Respiratory Infection When Novel Coronavirus (2019-nCoV). Infection Is Suspected: Interim Guidance. (2020). Available online at: <https://apps.who.int/iris/handle/10665/330893> (accessed April 31, 2020).
5. Zumla A, Hui DS, Perlman S. Middle East respiratory syndrome. *Lancet*. (2015) 386:995–1007. doi: 10.1016/S0140-6736(15)60454-8
6. Pham QV, Nguyen DC, Hwang WJ, Pathirana PN. Artificial intelligence (AI) and big data for coronavirus (COVID-19) pandemic: a survey on the state-of-the-arts. *Preprints*. (2020) 2020:2020040383. doi: 10.20944/preprints202004.0383.v1
7. WHO Situation Report-94 Coronavirus disease 2019 (COVID-19). (2020). Available online at: https://www.who.int/docs/default-source/coronaviruse/situation-reports/20200423-sitrep-94-covid-19.pdf?sfvrsn=b8304bf0_4 (accessed March 10, 2020).

8. Kathiresan S, Sait ARW, Gupta D, Lakshmanaprabu SK, Khanna A, Pandey HM. Automated detection and classification of fundus diabetic retinopathy images using synergic deep learning model. *Pattern Recogn Lett.* (2020) 133:210–6. doi: 10.1016/j.patrec.2020.02.026
9. Wang L, Wong A. COVID-Net: a tailored deep convolutional neural network design for detection of COVID-19 cases from chest radiography images. *arXiv.* (2020) 2003.09871. Available online at: <https://arxiv.org/abs/2003.09871> (accessed May 5, 2020).
10. Pal R, Sekh AA, Kar S, Prasad DK. Neural network-based country wise risk prediction of COVID-19. *arXiv.* (2020) 2004.00959. Available online at: <https://arxiv.org/abs/2004.00959> (accessed May 7, 2020).

Appendix of Code:

```
import warnings

warnings.filterwarnings('ignore')


import itertools

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from wordcloud import WordCloud

from sklearn import metrics


import re

import nltk

nltk.download('stopwords')

from nltk.corpus import stopwords

from sklearn.feature_extraction.text import TfidfVectorizer


from sklearn.tree import DecisionTreeClassifier

from sklearn.feature_extraction.text import CountVectorizer

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.cluster import KMeans

from sklearn.decomposition import PCA

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

from sklearn.model_selection import train_test_split

from sklearn.model_selection import cross_val_score
```

```

from sklearn.preprocessing import LabelEncoder

from sklearn.naive_bayes import MultinomialNB

data = pd.read_csv("Corona_NLP_train.csv",encoding='latin1')

df = pd.DataFrame(data)

df.head()

plt.figure(figsize=(10,5))

sns.countplot(x='Sentiment', data=df, order=['Extremely Negative', 'Negative', 'Neutral', 'Positive',
'Extremely Positive'], )

df.info()

df.head()

# Plot unique value across different colums

def unique_values(data_frame):

    unique_dataframe = pd.DataFrame()

    unique_dataframe['Features'] = data_frame.columns

    uniques = []

    for col in data_frame.columns:

        u = data_frame[col].nunique()

        uniques.append(u)

    unique_dataframe['Uniques'] = uniques

    return unique_dataframe

unique_data = unique_values(data)

f, ax = plt.subplots(1,1, figsize=(10,5))

sns.barplot(x=unique_data['Features'], y=unique_data['Uniques'], alpha=0.8)

plt.title('Bar plot for number of unique values in each column')

plt.ylabel('#Unique values', fontsize=12)

plt.xlabel('Features', fontsize=12)

plt.xticks(rotation=90)

plt.show()

```

```

all_words = ' '.join([text for text in df['CleanedTweet'] ])

wordcloud = WordCloud(width=800, height=500, random_state=21,
max_font_size=110).generate(all_words)

plt.figure(figsize=(10, 7))

plt.imshow(wordcloud, interpolation="bilinear")

plt.axis('off')

plt.show()

vectorizer = TfidfVectorizer(max_features=800)

tfidf_feature = vectorizer.fit_transform(df['CleanedTweet'].values)

kmeans_tfidf = KMeans(5)

y_pred_tfidf = kmeans_tfidf.fit_predict(tfidf_feature)

df['label_tfidf_kmeans'] = kmeans_tfidf.labels_

pca_tfidf = PCA(n_components=3).fit(tfidf_feature.toarray())

pca_tfidf_data = pca_tfidf.transform(tfidf_feature.toarray())

centroids_tfidf = pca_tfidf.transform(kmeans_tfidf.cluster_centers_)

sns.set(rc={'figure.figsize':(15,15)})

palette = sns.color_palette("bright", len(set(y_pred_tfidf)))

sns.scatterplot(pca_tfidf_data[:,0], pca_tfidf_data[:,1], hue=y_pred_tfidf, legend='full',
palette=palette)

plt.title("PCA Covid-19 Articles - Clustered (K-Means) - Tf-idf on cleaned abstract Text")

plt.show()

ax = plt.figure(figsize=(10,10)).gca(projection='3d')

ax.scatter(

    xs=pca_tfidf_data[:,0],

    ys=pca_tfidf_data[:,1],

    zs=pca_tfidf_data[:,2],

    c=y_pred_tfidf,

```

```

    cmap='tab10'
)
ax.set_xlabel('pca-axis-x')
ax.set_ylabel('pca-axis-y')
ax.set_zlabel('pca-axis-z')
plt.title("PCA Covid-19 Articles (3D) - Clustered (K-Means) - Tf-idf on cleaned Tweet Text")

plt.show()

ticklabel=tfidf_data.labels.unique()

_predictions = NB_tfidf_model.predict(X_tfidf_test)
print(_predictions)
_predictions_series = pd.Series(log_predictions)
_predictions_series = log_predictions_series.astype(int)

scores_multi = cross_val_score(DT_tfidf_model, X_tfidf_train, y_tfidf_train , cv=10)

print("Cross Validation Accuracy: %0.2f (+/- %0.2f)" % (scores_multi.mean(), scores_multi.std() *
2))

print("\n")

cm1=confusion_matrix(y_tfidf_test, _predictions_series, labels=ticklabel)

plt.figure()

plot_confusion_matrix(cm1, classes=ticklabel, title='DT model 5 classes prediction (tfidf feature)
results', fs=(12,12), font_size=16)

print(classification_report(y_tfidf_test, log_predictions_series))

```