SUMMER INTERNSHIP REPORT on

**CHATBOT** on **DATA SCIENCE**

Dissertation Submitted in partial fulfilment of the requirements of the

Award of the degree in

**Bachelor of Technology In Computer Science and Engineering**

**Submitted By**

**BANDI RAMA KRISHNA 20031A0508**

Under the esteemed guidance of

**Dr. G. MADHAVI**

Assistant Professor & HOD (I/C)

Department of CSE

UCEN-JNTUK

****

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**University College of Engineering Narasaraopet**

**Jawaharlal Nehru Technological University Kakinada**

**Narasaraopet – 522601, Andhra Pradesh, INDIA**

**2021 – 2022**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**

**University College of Engineering Narasaraopet**

**Jawaharlal Nehru Technological University Kakinada**

**Narasaraopet – 522601, Andhra Pradesh (2021 – 2022)**

****

**C E R T I F I C A T E**

This is to certify that the Technical report on Industrial Training entitled **“DATA SCIENCE”** that is being submitted by **BANDI RAMA KRISHNA (20031A0508)** in partial fulfilment for the award of Bachelor of Technology in Computer Science and Engineering to the University College of Engineering Narasaraopet, Jawaharlal Nehru Technological University Kakinada is a record of bonafide work carried out by her under our supervision.

This report has not been submitted to any other university/institute for the award of any degree.

**PROJECT SUPERVISOR HEAD OF THE DEPARTMENT**

**Dr. G. MADHAVI Dr. G. MADHAVI**

Assistant Professor & HOD (I/C) Assistant Professor & HOD (I/C)

Department of CSE Department of CSE

UCEN-JNTUK UCEN-JNTUK

**Signature of External Examiner**

**CERTIFICATE**

****

**DECLARATION**

We hereby declare that the work described in this project report entitled “**RULE BASED CHATBOT**” which is submitted by me in partial fulfillment for the award of the degree of **Bachelor of Technology** in the Department of **Computer Science And Engineering** to the college, University College of Engineering Narasaraopet, Jawaharlal Nehru Technological University Kakinada, Andhra Pradesh, is the result of work done by me under the supervision of **DR.G.MADHAVI**, Assistant Professor & HOD(I/C), Department of Computer Science and Engineering.

The work is original and has not been submitted for any Degree/Diploma of this or any other university.

**BANDI RAMA KRISHNA - 20031A0508 Signature**

**PLACE:** Narasaraopet

**DATE:**

**ACKNOWLEDGEMENT**

It is needed with a great sense of pleasure and immense sense of gratitude that we acknowledge the help of these individuals. We owe many thanks to many people who helped and supported us during the writing of this report. We take privilege to express our heartfelt gratitude to Project Supervisor, **DR.G.MADHAVI**, Assistant Professor & HOD ( I/C), and our Project Coordinator, **Mrs.B.JYOTHI**, Assistant professor of Department of CSE and our Head of the Department **DR.G.MADHAVI**, Assistant Professor & HOD ( I/C ) of Department of CSE for their valuable suggestions and constant motivation that greatly helped us in successful completion of the project. We express our sincere thanks to **Prof.Ch.Srinivasa Rao**, Principal, University College of Engineering Narasaraopet - JNTUK, for providing us with a good infrastructure and environment to carry out this project.

We are thankful to all faculty members for extending their kind cooperation and assistance. Finally, we are extremely thankful to our parents and friends for their constant help and moral support.

**BANDI RAMA KRISHNA – 20031A0508.**

**ABSTRACT**

Data science **encompasses a set of principles, problem definitions, algorithms, and processes for extracting nonobvious and useful patterns from large data sets**. Many of the elements of data science have been developed in related fields such as machine learning and data mining.

Traditionally, to get a question answered by a software program or search engine or filling out a form is a very hectic task if there are more requests for the frequently asked questions. So to overcome it chatbots are useful. The most well known chatbots currently are voice chatbots: Alexa and Siri. However chatbots are currently being adopted at a high rate in most computer platforms.

The major factors of motivation for designing chatbots are productivity, entertainment, social and relational factors, curiosity. Chatbots provide the assistance or access to information quickly and efficiently. Chatbots amuse people by giving them funny tips, they also help killing time when users have nothing to do.

Chatbot can be described as software that can chat with people using machine learning and AI. These software are used to perform tasks such as quickly responding to users, informing them in no time, helping to purchase products and providing better service to customers. The use of chatbots evolved rapidly in numerous fields in recent years, including Marketing, Supporting Systems, Education, Health Care, Cultural Heritage, and Entertainment.

Chatbots are important for companies since this project can answer all the questions posed by the clients and information without the process being slowing down. The customer support workload has been decreased by the procedures which is fully automating. This process can be easily obtained by implementing Machine Learning, Artificial Intelligence and **Data Science** techniques. Chatbots operate by assessing the customer‘s input and responding with a mapped response. Recurrent Neural Networks using the intentions JSON dataset may be used to train the chatbot, while Python can be used to implement it. The objective of the chatbot will determine whether it is domain-specific or open-domain.

**Table of Content**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **CHAPTER NAME** | **PG.NO** |
|  | **LIST OF FIGURES** |  |
|  | **LIST OF ABBREVIATIONS** |  |
| **1.** | **INTRODUCTION** |  |
|  | 1.1 INTRODUCTION |  |
|  | 1.2 SIGNIFICANCE OF DATA SCIENCE PROJECT |  |
|  | 1.2.1 DATA SCIENCE AND ITS IMPORTANCE |  |
|  | 1.2.2 HISTORY OF DATA SCIENCE |  |
|  | 1.2.3 SCOPE OF DATA SCIENCE |  |
| **2.** | **PROGRAMMING LANGUAGE** |  |
|  | 2.1 PYTHON |  |
|  | 2.1.1 SIGNIFICANCE OF PYTHON |  |
|  | 2.1.2 PYTHON FEATURES |  |
|  | 2.1.3 DATA STRUCTURES IN PYTHON |  |
|  | 2.2 NUMPY |  |
|  | 2.2.1 NUMPY N-DIMENSIONAL ARRAY |  |
|  | 2.2.2 PYTHON STATISTICS MODULE |  |
|  | 2.2.3 PYTHON MATPLOTLIB |  |
| **3.** | **MACHINE LEARNING** |  |
|  | 3.1 ROLE OF MACHINE LEARNING |  |
|  | 3.2 MACHINE LEARNING LIFE CYCLE |  |
|  | 3.3 THREE KEY MACHINE LEARNING ALGORITHMS |  |
|  | 3.4 DATASETS OF MACHINE LEARNING |  |
|  | **MINI PROJECT** |  |
| **4.** | **MINI PROJECT ON DATA SCIENCE** |  |
|  | 4.1 PROJECT NAME |  |
|  | 4.2 ROLE OF PROJECT |  |
|  | 4.3 USES OF PROJECT |  |
|  | 4.4 SCOPE OF PROJECT |  |
| **5.** | **SYSTEM REQUIREMENT SPECIFICATIONS** |  |
|  | 5.1 SOFTWARE REQUIREMENTS |  |
|  | 5.2 HARDWARE REQUIREMENTS |  |
| **6.** | **SYSTEM DESIGN** |  |
|  | 6.1 SYSTEM ARCHITECTURE |  |
|  | 6.2 UML DIAGRAMS USED IN DESIGN |  |
|  | 6.2.1 USE CASE DIAGRAM |  |
|  | 6.2.2 CLASS DIAGRAM |  |
|  | 6.2.3 SEQUENCE DIAGRAM |  |
|  | 6.2.4 ACTIVITY DIAGRAM |  |
|  | 6.2.5 DATAFLOW DIAGRAM |  |
| **7.** | **IMPLEMENTATION OF MINI PROJECT** |  |
|  | 7.1 SOURCE CODE OF MINI PROJECT |  |
|  | 7.2 DEBUGGING AND EXECUTING MINI PROJECT |  |
| **8.** | **RESULTS** |  |
|  | 8.1 SCREENSHOT-1 |  |
|  | 8.2 SCREENSHOT-2 |  |
| **9.** | **CONCLUSION** |  |
| **10.** | **REFERENCES** |  |

**LIST OF FIGURES**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **FIG NO** | **DESCRIPTION** | **Pg.NO** |
| 1 | 2.1.3 | Flowchart of Data Structures in Python |  |
| 2 | 2.1.3.1 | Stack representation |  |
| 3 | 2.1.3.2 | Queue representation |  |
| 4 | 2.1.3.3 | Tree representation |  |
| 5 | 2.1.3.4 | Linked list representation |  |
| 6 | 2.1.3.5 | Graph representation |  |
| 7 | 2.1.3.6 | Hash map representation |  |
| 8 | 2.2.3.1 | Line plot output |  |
| 9 | 2.2.3.2 | Bar plot output |  |
| 10 | 2.2.3.3 | Histogram output |  |
| 11 | 2.2.3.4 | Scanner plot output |  |
| 12 | 3.1 | Machine learning roles |  |
| 13 | 3.3.1 | Simple linear regression graph |  |
| 14 | 3.3.2 | Classification graph |  |
| 15 | 3.3.3 | Clustering graph |  |
| 16 | 3.4 | Datasets representation |  |
| 17 | 6.1 | System Architecture |  |
| 18 | 8.1 | Output of ChatBot |  |
| 19 | 8.2 | Conversation with ChatBot |  |

**LIST OF ABBREVIATIONS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **ABBREVIATION** | **FULL FORM** |
| 1 | AI | Artificial Intelligence |
| 2 | ML | Machine Learning |
| 3 | GUI | Graphical User Interface |
| 4 | NLP | Natural Language Processing |
| 5 | OOP | Object Oriented Programming |
| 6 | DFD | Data Flow Diagram |
| 7 | OS | Operating System |

**INTRODUCTION**

1. **INTRODUCTION**
   1. **INTRODUCTION**

**Data Science** is a combination of multiple disciplines that uses statistics, data analysis, and machine learning to analyze data and to extract knowledge and insights from it. Data Science is about data gathering, analysis and decision-making. Data Science is about finding patterns in data, through analysis, and make future predictions. Data Science is used in many industries in the world today, e.g. banking, consultancy, healthcare, and manufacturing.

Data Science is kind a blended with various tools, algorithms, and machine learning principles. Most simply, it involves obtaining meaningful information or insights from structured or unstructured data through a process of analyzing, programming and business skills. It is a field containing many elements like mathematics, statistics, computer science, etc. Those who are good at these respective fields with enough knowledge of the domain in which you are willing to work can call themselves as Data Scientist. It’s not an easy thing to do but not impossible too. You need to start from data, it’s visualization, programming, formulation, development, and deployment of your model. In the future, there will be great hype for data scientist jobs. Taking in that mind, be ready to prepare yourself to fit in this world.

* 1. **Significance of Data Science**
     1. **What Is Data Science & It’s Importance**

Data is meaningless until its conversion into valuable information. Data Science involves mining large datasets containing structured and unstructured data and identifying hidden patterns to extract actionable insights. The importance of Data Science lies in its innumerable uses that range from daily activities like asking Siri or Alexa for recommendations to more complex applications like operating a self-driving car.

The interdisciplinary field of Data Science encompasses Computer Science, Statistics, Inference, Machine Learning algorithms, Predictive Analysis, and new technologies.

Data is a precious asset of any organization. It helps firms understand and enhance their processes, thereby saving time and money. Wastage of time and money, such as a terrible advertising decision, can deplete resources and severely impact a business. The efficient use of data enables businesses to reduce such wastage by analyzing different marketing channels’ performance and focusing on those offering the highest ROI. Thus, a company can generate more leads without increasing its advertising spend.

* + 1. **History Of Data Science**

In 1962, John Tukey wrote about the convergence of Statistics and computers to devise measurable outputs in hours. In 1974, Peter Naur mentioned the term ‘Data Science’ multiple times in his review, Concise Survey of Computer Methods. In 1977, the International Association for Statistical Computing (IASC) was formed to link modern computer technology, traditional statistical methodology, and domain expertise to convert data into knowledge. In the same year, Tukey composed a paper, Exploratory Data Analysis, that briefed the importance of using data.

By 1994, organizations had started gathering tremendous individual data for new showcasing efforts. In 1999, Jacob Zahavi stressed the need for new devices to deal with the gigantic chunk of organizational data. In 2001, William S. Cleveland presented an activity plan depicting how to create a specialized understanding and scope of Data Scientists and indicated six regions of studies for offices and colleges.

In 2002, the International Council for Science published the Data Science Journal focusing on Data Science issues like data systems explanation, application, and more. In 2003, Columbia University published the Data Science Journal to set a platform for data teams. In the year 2005, the National Science Board published an existing collection of digital data, and in 2013, IBM revealed that 90% of the global data had been created in the past two years. By this time, organizations realized the importance of Data Science to convert huge data clusters into usable information to gain crucial insights.

**1.2.3 Scope of Data Science:**

Especially in a developing country like India, there is scope for huge data-related operations such as **data scientists, data analytics, big data engineers, big data managers, and data architects**

**PROGRAMMING LANGUAGES**

1. **PROGRAMMING LANGUAES**
   1. **PYTHON**
      1. **Significance of Python**

This data science with Python tutorial will help you learn the basics of Python along with different steps of data science such as data preprocessing, data visualization, statistics, making machine learning models, and much more with the help of detailed and well-explained examples. This tutorial will help both beginners as well as some trained professionals in mastering data science with Python.

# **2.1.2 Python Features:**

Python provides many useful features which make it popular and valuable from the other programming languages. It supports object-oriented programming, procedural programming approaches and provides dynamic memory allocation. We have listed below a few essential features.

### **1) Easy to Learn and Use**

Python is easy to learn as compared to other programming languages. Its syntax is straightforward and much the same as the English language. There is no use of the semicolon or curly-bracket, the indentation defines the code block. It is the recommended programming language for beginners.

### **2) Expressive Language**

Python can perform complex tasks using a few lines of code. A simple example, the hello world program you simply type **print("Hello World")**. It will take only one line to execute, while Java or C takes multiple lines.

### **3) Interpreted Language**

Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.

### **4) Cross-platform Language**

Python can run equally on different platforms such as Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a portable language. It enables programmers to develop the software for several competing platforms by writing a program only once.

### **5) Free and Open Source**

Python is freely available for everyone. It is freely available on its official website [www.python.org](https://www.python.org/). It has a large community across the world that is dedicatedly working towards make new python modules and functions. Anyone can contribute to the Python community. The open-source means, "Anyone can download its source code without paying any penny."

### **6) Object-Oriented Language**

Python supports object-oriented language and concepts of classes and objects come into existence. It supports inheritance, polymorphism, and encapsulation, etc. The object-oriented procedure helps to programmer to write reusable code and develop applications in less code.

### **7) Extensible**

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our Python code. It converts the program into byte code, and any platform can use that byte code.

### **8) Large Standard Library**

It provides a vast range of libraries for the various fields such as machine learning, web developer, and also for the scripting. There are various machine learning libraries, such as Tensor flow, Pandas, Numpy, Keras, and Pytorch, etc. Django, flask, pyramids are the popular framework for Python web development.

### **9) GUI Programming Support**

Graphical User Interface is used for the developing Desktop application. PyQT5, Tkinter, Kivy are the libraries which are used for developing the web application.

### **10) Integrated**

It can be easily integrated with languages like C, C++, and JAVA, etc. Python runs code line by line like C,C++ Java. It makes easy to debug the code.

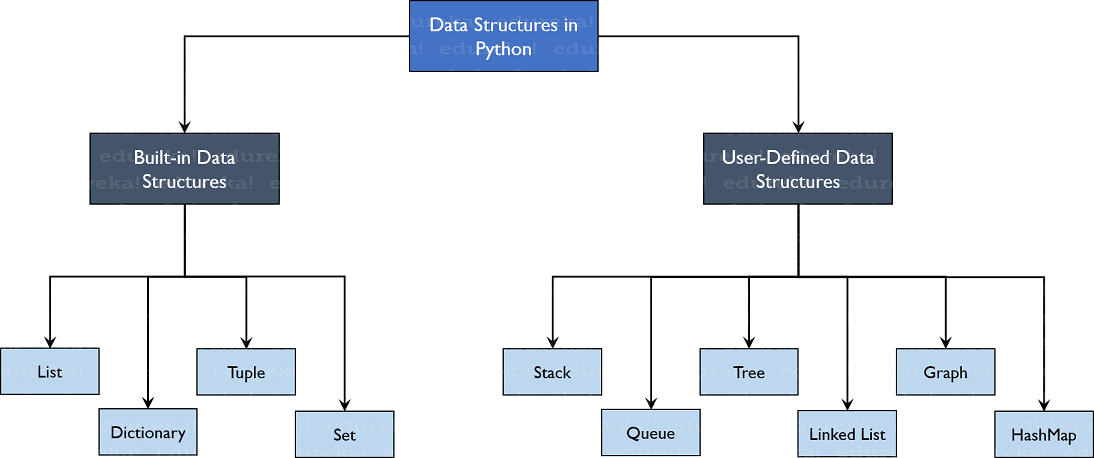
### **11. Embeddable**

The code of the other programming language can use in the Python source code. We can use Python source code in another programming language as well. It can embed other language into our code.

### **12. Dynamic Memory Allocation**

In Python, we don't need to specify the data-type of the variable. When we assign some value to the variable, it automatically allocates the memory to the variable at run time. Suppose we are assigned integer value 15 to **x,** then we don't need to write **int x = 15.** Just write x = 15.

## Data Structures in Python

Python has **implicit** support for Data Structures which enable you to store and access data. These structures are called [List](https://www.edureka.co/blog/lists-in-python/), [Dictionary](https://www.edureka.co/blog/dictionary-in-python/), [Tuple](https://www.edureka.co/blog/tuple-in-python/) and [Set](https://www.edureka.co/blog/sets-in-python/).

2.1.3 Flowchart of Data Structures in Python

Python allows its users to create their own Data Structures enabling them to have **full control** over their [functionality](https://www.edureka.co/blog/python-functions). The most prominent Data Structures are Stack, Queue, Tree, Linked List and so on which are also available to you in other programming languages. So now that you know what are the types available to you, why don’t we move ahead to the Data Structures and implement them using Python.

## ****Built-in Data Structures****

As the name suggests, these Data Structures are built-in with [Python which makes programming easier](https://www.edureka.co/blog/python-basics/) and helps programmers use them to obtain solutions faster. Let’s discuss each of them in detail.

### **Lists**

[Lists](https://www.edureka.co/blog/lists-in-python/) are used to store data of different data types in a sequential manner. There are addresses assigned to every element of the list, which is called as Index. The index value starts from 0 and goes on until the last element called the **positive index**. There is also **negative indexing** which starts from -1 enabling you to access elements from the last to first. Let us now understand lists better with the help of an example program.

#### **Creating a list**

|  |  |
| --- | --- |
| 1  2  3  4 | my\_list = [] #create empty list  print(my\_list)  my\_list = [1, 2, 3, 'example', 3.132] #creating list with data  print(my\_list) |

To create a list, you use the square brackets and add elements into it accordingly. If you do not pass any elements inside the square brackets, you get an empty list as the output.

**Output:**  
[]  
[1, 2, 3, ‘example’, 3.132]

### **Dictionary**

[Dictionaries](https://www.edureka.co/blog/dictionary-in-python/) are used to store **key-value** pairs. To understand better, think of a phone directory where hundreds and thousands of names and their corresponding numbers have been added. Now the constant values here are Name and the Phone Numbers which are called as the keys. And the various names and phone numbers are the values that have been fed to the keys. If you access the values of the keys, you will obtain all the names and phone numbers. So that is what a key-value pair is. And in Python, this structure is stored using Dictionaries. Let us understand this better with an example program.

#### **Creating a Dictionary**

Dictionaries can be created using the flower braces or using the dict() function. You need to add the key-value pairs whenever you work with dictionaries.

|  |  |
| --- | --- |
| 1  2  3  4 | my\_dict = {} #empty dictionary  print(my\_dict)  my\_dict = {1: 'Python', 2: 'Java'} #dictionary with elements  print(my\_dict) |

**Output:**  
{}  
{1: ‘Python’, 2: ‘Java’}

### **Tuple**

[Tuples](https://www.edureka.co/blog/tuple-in-python/) are the same as lists are with the exception that the data once entered into the tuple cannot be changed no matter what. The only exception is when the data inside the tuple is mutable, only then the tuple data can be changed. The example program will help you understand better.

#### **Creating a Tuple**

You create a tuple using parenthesis or using the tuple() function.

|  |  |
| --- | --- |
| 1  2 | my\_tuple = (1, 2, 3) #create tuple  print(my\_tuple) |

**Output:**  
(1, 2, 3)

### **Sets**

[Sets](https://www.edureka.co/blog/sets-in-python/) are a collection of unordered elements that are unique. Meaning that even if the data is repeated more than one time, it would be entered into the set only once. It resembles the sets that you have learnt in arithmetic. The operations also are the same as is with the arithmetic sets. An example program would help you understand better.

#### **Creating a set**

Sets are created using the flower braces but instead of adding key-value pairs, you just pass values to it.

|  |  |
| --- | --- |
| 1  2 | my\_set = {1, 2, 3, 4, 5, 5, 5} #create set  print(my\_set) |

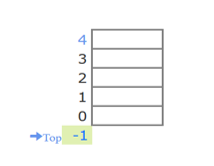
**Output:**  
{1, 2, 3, 4, 5}

## ****User-Defined Data Structures****

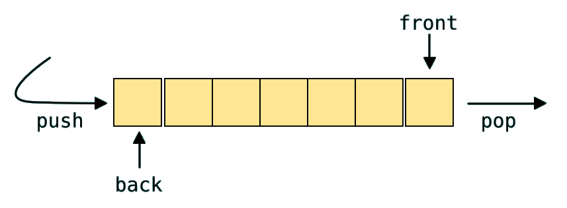
### **Arrays vs. Lists**

Arrays and lists are the same structure with one difference. Lists allow heterogeneous data element storage whereas [Arrays](https://www.edureka.co/blog/arrays-in-python/) allow only homogenous elements to be stored within them.

### **Stack**

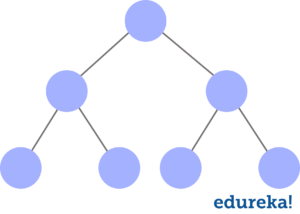
[Stacks](https://www.edureka.co/blog/stack-in-python/) are linear Data Structures which are based on the principle of Last-In-First-Out (LIFO) where data which is entered last will be the first to get accessed. It is built using the array structure and has operations namely, pushing (adding) elements, popping (deleting) elements and accessing elements only from one point in the stack called as the TOP. This TOP is the pointer to the current position of the stack. Stacks are prominently used in applications such as Recursive Programming, reversing words, undo mechanisms in word editors and so forth. 2.1.3.1 Stack representation

### **Queue**

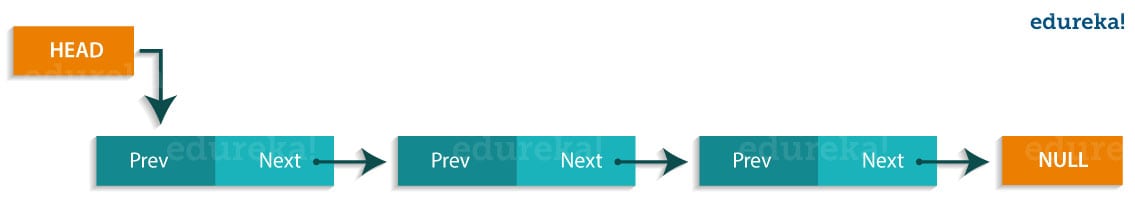
A [queue](https://www.edureka.co/blog/queue-data-structure-in-python/) is also a linear data structure which is based on the principle of First-In-First-Out (FIFO) where the data entered first will be accessed first. It is built using the array structure and has operations which can be performed from both ends of the Queue, that is, head-tail or front-back. Operations such as adding and deleting elements are called En-Queue and De-Queue and accessing the elements can be performed. Queues are used as Network Buffers for traffic congestion management, used in Operating Systems for Job Scheduling and many more.

2.1.3.2 Queue representation

### **Tree**

Trees are non-linear Data Structures which have root and nodes. The root is the node from where the data originates and the nodes are the other data points that are available to us. The node that precedes is the parent and the node after is called the child. There are levels a tree has to show the depth of information. The last nodes are called the leaves. Trees create a hierarchy which can be used in a lot of real-world applications such as the [HTML](https://www.edureka.co/blog/what-is-html/) pages use trees to distinguish which tag comes under which block. It is also efficient in searching purposes and much more. 2.1.3.3 Tree representation

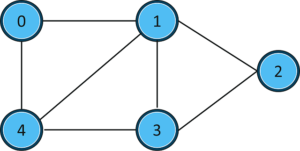
### **Linked List**

[Linked lists](https://www.edureka.co/blog/linked-list-in-python/) are linear Data Structures which are not stored consequently but are linked with each other using pointers. The node of a linked list is composed of data and a pointer called next. These structures are most widely used in image viewing applications, music player applications and so forth.

2.1.3.4 Linked list representation

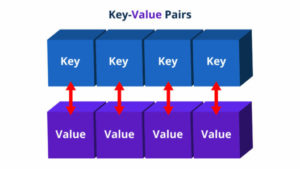
**Graph**

Graphs are used to store data collection of points called vertices (nodes) and edges (edges). Graphs can be called as the most accurate representation of a real-world map. They are used to find the various cost-to-distance between the various data points called as the nodes and hence find the least path. Many applications such as Google Maps, Uber, and many more use Graphs to find the least distance and increase profits in the best ways.

  
2.1.3.5 Graph representation

### **HashMaps**

[HashMaps](https://www.edureka.co/blog/dictionary-in-python/) are the same as what dictionaries are in Python. They can be used to implement applications such as phonebooks, populate data according to the lists and much more.



2.1.3.6 Hashmap representation

* 1. **NumPy**

Arrays are the main data structure used in machine learning.

In Python, arrays from the NumPy library, called [N-dimensional arrays or the ndarray](https://machinelearningmastery.com/index-slice-reshape-numpy-arrays-machine-learning-python/), are used as the primary data structure for representing data.

In this tutorial, you will discover the N-dimensional array in NumPy for representing numerical and manipulating data in Python.

**2.2.1 NumPy N-dimensional Array**

NumPy is a Python library that can be used for scientific and numerical applications and is the tool to use for linear algebra operations.

The main data structure in NumPy is the ndarray, which is a shorthand name for N-dimensional array. When working with NumPy, data in an ndarray is simply referred to as an array.

It is a fixed-sized array in memory that contains data of the same type, such as integers or floating point values.

The data type supported by an array can be accessed via the “dtype” attribute on the array. The dimensions of an array can be accessed via the “shape” attribute that returns a tuple describing the length of each dimension. There are a host of other attributes. Learn more here:

* [**The N-dimensional array**](https://docs.scipy.org/doc/numpy-1.13.0/reference/arrays.ndarray.html)

A simple way to create an array from data or simple Python data structures like a list is to use the array() function.

The example below creates a Python list of 3 floating point values, then creates an ndarray from the list and access the arrays’ shape and data type.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | # create array  from numpy import array  l = [1.0, 2.0, 3.0]  a = array(l)  print(a)  print(a.shape)  print(a.dtype) |

Running the example prints the contents of the ndarray, the shape, which is a one-dimensional array with 3 elements, and the data type, which is a 64-bit floating point.

|  |  |
| --- | --- |
| 1  2  3 | [ 1. 2. 3.]  (3,)  float64 |

* + 1. **Python statistics Module**

Python has a built-in module that you can use to calculate mathematical statistics of numeric data.

The statistics module was new in Python 3.4.

**Statistics Methods**

In the era of big data and [artificial intelligence](https://realpython.com/python-ai-neural-network/), [data science](https://realpython.com/tutorials/data-science/) and [machine learning](https://realpython.com/tutorials/machine-learning/) have become essential in many fields of science and technology. A necessary aspect of working with data is the ability to describe, summarize, and represent data visually.

**Python statistics libraries** are comprehensive, popular, and widely used tools that will assist you in working with data.

|  |  |
| --- | --- |
| **Method** | **Description** |
| [statistics.harmonic\_mean()](https://www.w3schools.com/python/ref_stat_harmonic_mean.asp) | Calculates the harmonic mean (central location) of the given data |
| [statistics.mean()](https://www.w3schools.com/python/ref_stat_mean.asp) | Calculates the mean (average) of the given data |
| [statistics.median()](https://www.w3schools.com/python/ref_stat_median.asp) | Calculates the median (middle value) of the given data |
| [statistics.median\_grouped()](https://www.w3schools.com/python/ref_stat_median_grouped.asp) | Calculates the median of grouped continuous data |
| [statistics.median\_high()](https://www.w3schools.com/python/ref_stat_median_high.asp) | Calculates the high median of the given data |
| [statistics.median\_low()](https://www.w3schools.com/python/ref_stat_median_low.asp) | Calculates the low median of the given data |
| [statistics.mode()](https://www.w3schools.com/python/ref_stat_mode.asp) | Calculates the mode (central tendency) of the given numeric or nominal data |
| [statistics.pstdev()](https://www.w3schools.com/python/ref_stat_pstdev.asp) | Calculates the standard deviation from an entire population |
| [statistics.stdev()](https://www.w3schools.com/python/ref_stat_stdev.asp) | Calculates the standard deviation from a sample of data |
| [statistics.pvariance()](https://www.w3schools.com/python/ref_stat_pvariance.asp) | Calculates the variance of an entire population |
| [statistics.variance()](https://www.w3schools.com/python/ref_stat_variance.asp) | Calculates the variance from a sample of data |

# **2.2.3 Python Matplotlib**

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

**Installation:**

Windows, Linux and mac OS distributions have matplotlib and most of its dependencies as wheel packages. Run the following command to install matplotlib package :

python -mpip install -U matplotlib

**Importing matplotlib :**

from matplotlib import pyplot as plt

or

import matplotlib.pyplot as plt

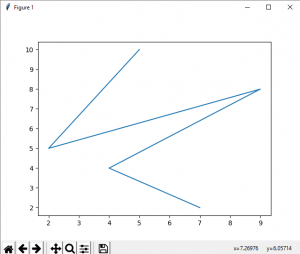
#### Basic plots in Matplotlib :

Matplotlib comes with a wide variety of plots. Plots helps to understand trends, patterns, and to make correlations. They’re typically instruments for reasoning about quantitative information. Some of the sample plots are covered here.

**Line plot :**

|  |
| --- |
| # importing matplotlib module  from matplotlib import pyplot as plt  # x-axis values  x = [5, 2, 9, 4, 7]  # Y-axis values  y = [10, 5, 8, 4, 2]  # Function to plot  plt.plot(x,y)  # function to show the plot  plt.show() |

**Output :**

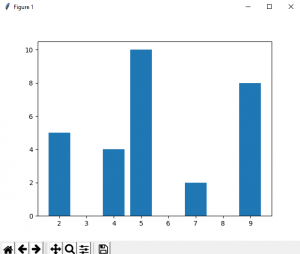


**2.2.3.1 Line plot output**

**Bar plot :**

|  |
| --- |
| # importing matplotlib module  from matplotlib import pyplot as plt  # x-axis values  x = [5, 2, 9, 4, 7]  # Y-axis values  y = [10, 5, 8, 4, 2]  # Function to plot the bar  plt.bar(x,y)  # function to show the plot  plt.show() |

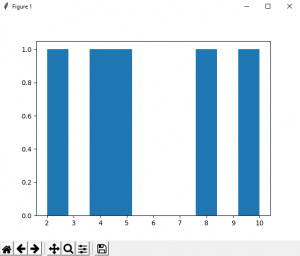
**Output :**

  
2.2.3.2 Barplot output

**Histogram :**

|  |
| --- |
| # importing matplotlib module  from matplotlib import pyplot as plt  # Y-axis values  y = [10, 5, 8, 4, 2]  # Function to plot histogram  plt.hist(y)  # Function to show the plot  plt.show() |

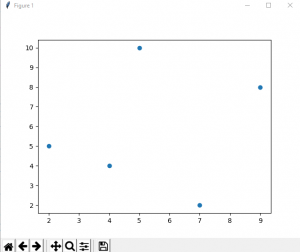
**Output:**



**2.2.3.3 Histogram output**

**Scatter Plot :**

|  |
| --- |
| # importing matplotlib module  from matplotlib import pyplot as plt  # x-axis values  x = [5, 2, 9, 4, 7]  # Y-axis values  y = [10, 5, 8, 4, 2]  # Function to plot scatter  plt.scatter(x, y)  # function to show the plot  plt.show() |

**Output:**

2.2.3.4 Scanner plot output

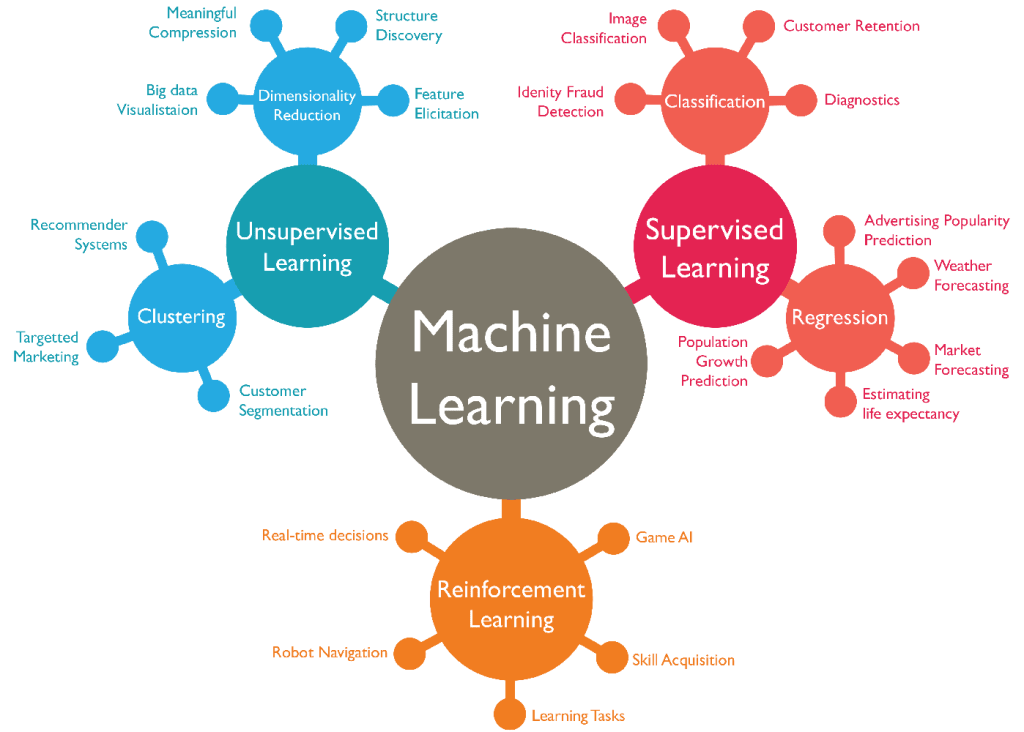
**MACHINE LEARNING**

1. **MACHINE LEARNING**

## Role of Machine Learning

It is now possible to Train Machines with a **Data-Driven** approach. On a wider spectrum, if you think of Artificial Intelligence as the main umbrella, Machine Learning is a subset of Artificial Intelligence. Machine Learning, a set of Algorithms, gives Machines or Computers the ability to learn from data on their own without any human intervention.

The idea behind Machine Learning is that you teach and Train Machines by feeding them data and defining features. Computers **learn, grow, adapt, and develop** by themselves when they are fed with new and relevant data, without relying on explicit programming. Without data, there is very little that Machines can learn. The Machine observes the dataset, identifies patterns in it, learns automatically from the behavior, and makes predictions.



3.1 Machine learning roles

It is the Machine Learning technology that Online Recommendation Engines use to offer relevant recommendations to the user, be it **YouTube Video Recommendations** or **Facebook Friend Recommendations**. One of the most recent technologies, **Google’s Self Driving Car** also makes use of Machine Learning Algorithms to understand the patterns and definitions, learn automatically, and execute the operation.

## Machine Learning Lifecycle

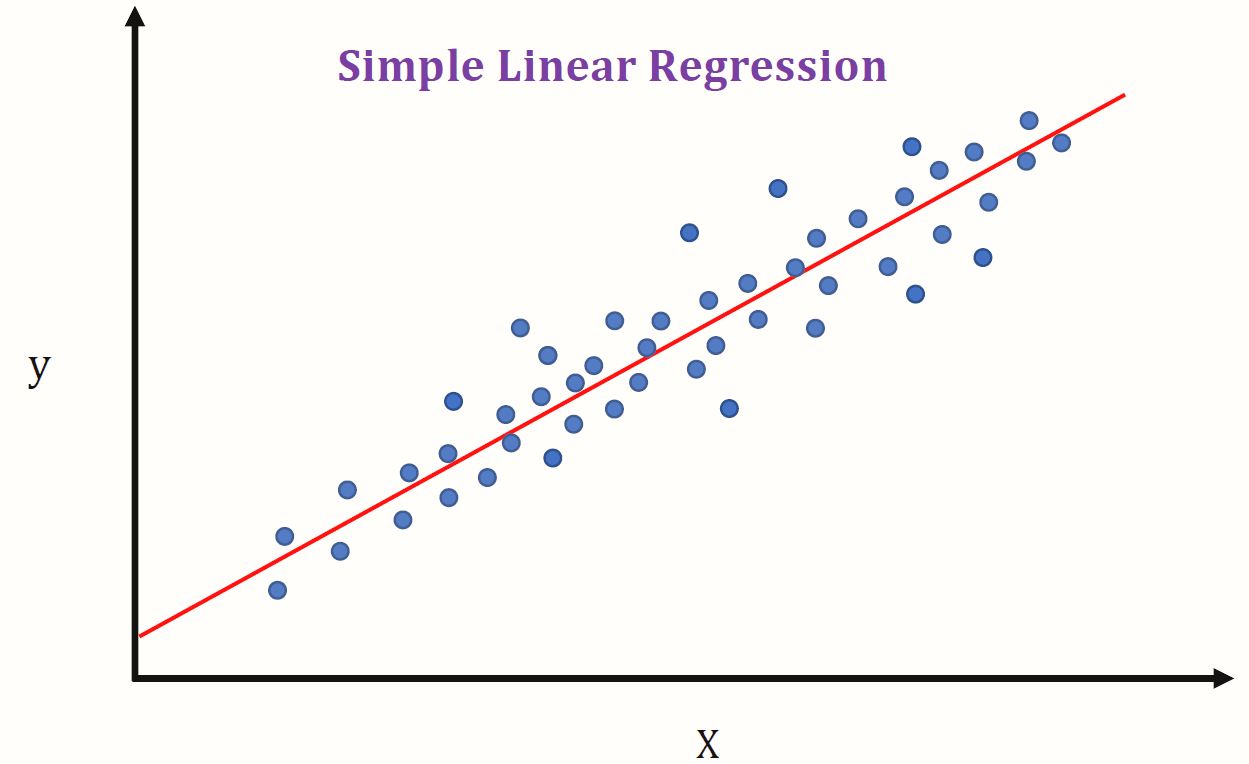
* **Data Collection:** Collecting data is considered the foundation step of Machine Learning. Collecting relevant and reliable data becomes very important as the quality and extent of data directly impact the outcome of your Machine Learning Model. As discussed in the previous section, this dataset is further used for Training your Data Model.
* **Data Preparation:** Data Cleaning is the first step in the overall Data Preparation process. This is an essential step in making the data analysis-ready. Data Preparation ensures that the dataset is free of erroneous or corrupt data points. It also involves standardizing the data into a single format. The dataset is also split into two parts to be used for Training your Data Model and evaluating the performance of the Trained Model, respectively.
* **Training the Model:** This is where the “learning” starts. The Training dataset is used to predict the output value. This output is bound to diverge from the desired value in the first iteration. But practice makes a “Machine” perfect. The step is repeated again after making some adjustments in the initialization. The Training data is used to incrementally improve the prediction accuracy of your Model.
* **Model Evaluation:** Once you’re done Training your Model, it’s time to evaluate its performance. The evaluation process makes use of the dataset that was set aside in the Data Preparation process. This data has never been used for Training the Model. So, Testing your Data Model against a new dataset will give you an idea of how your Model is going to perform in real-life applications.
* **Prediction:** Now that your Model is Trained and evaluated, it doesn’t mean that it’s perfect and is ready to be deployed. The Model is further improved by **tuning** the parameters. Prediction is the final step of Machine Learning. This is the step where your Data Model is deployed and the Machine makes use of its learning to answer your questions.

## Three Key Machine Learning Algorithms

When you have a dataset, you can classify the problem into three types:

* [Regression](https://hevodata.com/learn/machine-learning-in-data-science-2/#reg)
* [Classification](https://hevodata.com/learn/machine-learning-in-data-science-2/#cla)
* [Clustering](https://hevodata.com/learn/machine-learning-in-data-science-2/#clu)

### **1) Regression**

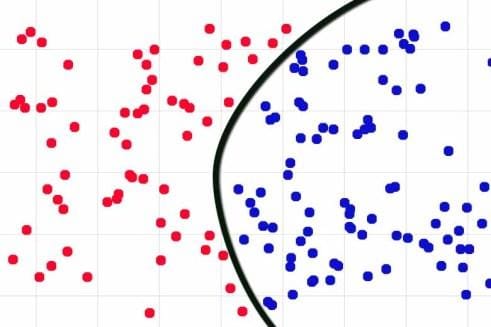
When the output variable is in continuous space, Regression is used. You probably would have come across the Curve-Fitting Techniques in Mathematics. “**y=mx+c**”, rings a bell? Regression, also, is based on the same techniques. Regression is more like finding the equation of a curve that fits the data points and once you have the equation, you can predict the output values accordingly.

3.3.1 Simple linear regression graph

Some famous Regression Algorithms are **Linear Regression**, **Perceptron**, and **Neural Networks**.

Regression is useful for Financial Predictions like Stock Market Prediction and Housing Price Prediction.

### **2) Classification**

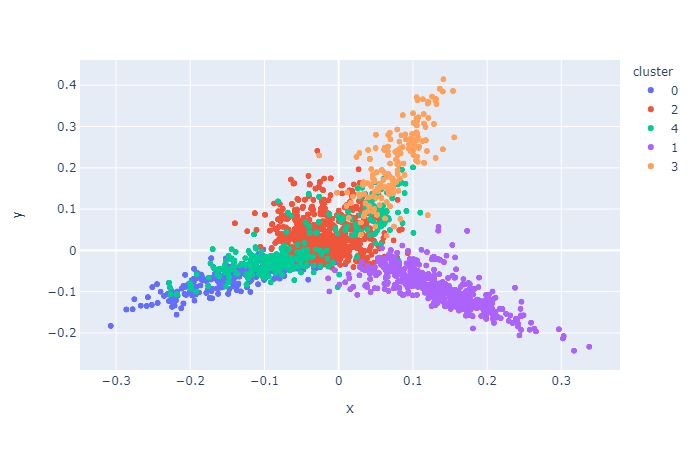
When the output variables are **discrete** values, Classification is used. If you want to find the category that your data belongs to, then it is a Classification problem. Classification Algorithms look at existing data to help you to predict the Class or Category of the new data.

Classification is more like finding curves that separate the data points into different Classes/Categories.

Labeling an Email as Spam is a Classification problem. Gmail, for example, will check any Email for the features that define Spam and will start putting it into your Spam Folder if 80% or 90% of the features match. 3.3.2 Classification graph

Some famous Classification Algorithms are **Support Vector Machines**, **Neural Networks**, **Naive Bayes**, **Logistic Regression**, and the **K Nearest Neighbour**.

### **3) Clustering**

If you just want to group your data points, having similar characteristics, without labels, it is then a Clustering problem. Ideally, the similar data points are grouped together in the same Cluster based on different definitions of similarity. The points in different Clusters should be as dissimilar as possible. The Clustering Algorithms try to find a pattern in a dataset without associating labels with it. 3.3.3 Clustering graph

Some famous Clustering Algorithms are **K-Means Clustering** and **Agglomerative Clustering**.

Buying behaviour of customers is Clustered using this Algorithm.

Regression and Classification come under the [Supervised Learning Model](https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning) of Machine Learning while Clustering comes under the Unsupervised Learning Model.

* 1. **DATASETS**

**A dataset** is a collection of data in which data is arranged in some order. A dataset can contain any data from a series of an array to a database table. Below table shows an example of the dataset:

|  |  |  |  |
| --- | --- | --- | --- |
| **Country** | **Age** | **Salary** | **Purchased** |
| India | 38 | 48000 | No |
| France | 43 | 45000 | Yes |
| Germany | 30 | 54000 | No |
| France | 48 | 65000 | No |
| Germany | 40 |  | Yes |
| India | 35 | 58000 | Yes |

A tabular dataset can be understood as a database table or matrix, where each column corresponds to a **particular variable,** and each row corresponds to the **fields of the dataset.** The most supported file type for a tabular dataset is **"Comma Separated File,"** or **CSV.** But to store a "tree-like data," we can use the JSON file more efficiently.

## Types of data in datasets

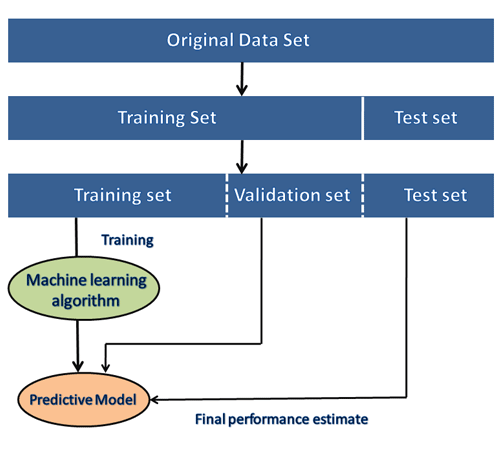
* **Numerical data:** Such as house price, temperature, etc.
* **Categorical data:** Such as Yes/No, True/False, Blue/green, etc.
* **Ordinal data:** These data are similar to categorical data but can be measured on the basis of comparison.

#### **Note:** A real-world dataset is of huge size, which is difficult to manage and process at the initial level. Therefore, to practice machine learning algorithms, we can use any dummy dataset.

## Need of Dataset

To work with machine learning projects, we need a huge amount of data, because, without the data, one cannot train ML/AI models. Collecting and preparing the dataset is one of the most crucial parts while creating an ML/AI project.

The technology applied behind any ML projects cannot work properly if the dataset is not well prepared and pre-processed. During the development of the ML project, the developers completely rely on the datasets. In building ML applications, datasets are divided into two parts:

* **Training dataset**
* **Test Dataset**

3.4 Datasets representation

#### **Note: The datasets are of large size, so to download these datasets, you must have fast internet on your computer.**

## Popular sources for Machine Learning datasets

Below is the list of datasets which are freely available for the public to work on it:

### **1. Kaggle Datasets**

Kaggle is one of the best sources for providing datasets for Data Scientists and Machine Learners. It allows users to find, download, and publish datasets in an easy way. It also provides the opportunity to work with other machine learning engineers and solve difficult Data Science related tasks.

Kaggle provides a high-quality dataset in different formats that we can easily find and download.

The link for the Kaggle dataset is <https://www.kaggle.com/datasets>.

### **2. UCI Machine Learning Repository**

UCI Machine learning repository is one of the great sources of machine learning datasets. This repository contains databases, domain theories, and data generators that are widely used by the machine learning community for the analysis of ML algorithms.

Since the year 1987, it has been widely used by students, professors, researchers as a primary source of machine learning dataset.

It classifies the datasets as per the problems and tasks of machine learning such as **Regression, Classification, Clustering, etc.** It also contains some of the popular datasets such as the **Iris dataset, Car Evaluation dataset, Poker Hand dataset, etc.**

The link for the UCI machine learning repository <https://archive.ics.uci.edu/ml/index.php>.

### **3. Datasets via AWS**

We can search, download, access, and share the datasets that are publicly available via AWS resources. These datasets can be accessed through AWS resources but provided and maintained by different government organizations, researches, businesses, or individuals.

Anyone can analyze and build various services using shared data via AWS resources. The shared dataset on cloud helps users to spend more time on data analysis rather than on acquisitions of data.

This source provides the various types of datasets with examples and ways to use the dataset. It also provides the search box using which we can search for the required dataset. Anyone can add any dataset or example to the **Registry of Open Data on AWS.**

The link for the resource is <https://registry.opendata.aws/>.

### **4. Google's Dataset Search Engine**

**Google dataset search engine** is a search engine launched by **Google** on **September 5, 2018.** This source helps researchers to get online datasets that are freely available for use.

The link for the Google dataset search engine is <https://toolbox.google.com/datasetsearch>.

### **5. Microsoft Datasets**

The Microsoft has launched the **"Microsoft Research Open data"** repository with the collection of free datasets in various areas such as **natural language processing, computer vision, and domain-specific sciences.**

Using this resource, we can download the datasets to use on the current device, or we can also directly use it on the cloud infrastructure.

The link to download or use the dataset from this resource is <https://msropendata.com/>.

### **6. Awesome Public Dataset Collection**

Awesome public dataset collection provides high-quality datasets that are arranged in a well-organized manner within a list according to topics such as Agriculture, Biology, Climate, Complex networks, etc. Most of the datasets are available free, but some may not, so it is better to check the license before downloading the dataset.

The link to download the dataset from Awesome public dataset collection is <https://github.com/awesomedata/awesome-public-datasets>.

### **7. Government Datasets**

There are different sources to get government-related data. Various countries publish government data for public use collected by them from different departments.

The goal of providing these datasets is to increase transparency of government work among the people and to use the data in an innovative approach. Below are some links of government datasets:

* [Indian Government dataset](https://data.gov.in/)
* [US Government Dataset](https://www.data.gov/)
* [Northern Ireland Public Sector Datasets](https://www.opendatani.gov.uk/)
* [European Union Open Data Portal](https://data.europa.eu/euodp/data/dataset)

### **8. Computer Vision Datasets**

Visual data provides multiple numbers of the great dataset that are specific to computer visions such as Image Classification, Video classification, Image Segmentation, etc. Therefore, if you want to build a project on deep learning or image processing, then you can refer to this source.

The link for downloading the dataset from this source is <https://www.visualdata.io/>.

### **9. Scikit-learn dataset**

Scikit-learn is a great source for machine learning enthusiasts. This source provides both toy and real-world datasets. These datasets can be obtained from sklearn.datasets package and using general dataset API.

The toy dataset available on scikit-learn can be loaded using some predefined functions such as, **load\_boston([return\_X\_y]), load\_iris([return\_X\_y]),** etc, rather than importing any file from external sources. But these datasets are not suitable for real-world projects.

**=**

**MINI PROJECT ON DATA SCIENCE**

1. **MINI PROJECT ON DATA SCIENCE**

**4.1 Project Name: CHATBOT**

A computer program designed to simulate conversation with human users, especially over the internet.

**4.2 Role of CHATBOT**

Chatbots present a new way for individuals to interact with computer systems or websites etc… The technology at the core of the rise of chatbots is natural language processing (NLP). Recent advances in Machine Learning which is a part in data science have greatly improved accuracy and effectiveness of NLP, Making chatbots a viable option for many organizations. A simple chatbots can be created by loading FAQs (Frequently Asked Questions) into chatbots software. The functionality of the chatbots can be improved by integrating it into the organization‘s enterprise software.

A chatbot is a program that communicates with you. It is a layer on top of, or a gateway to, a service. Sometimes it is powered by machine learning (the chatbot gets smarter the more you interact with it). Or, more commonly, it is driven using intelligent rules (i.e. if the person says this, respond with that).

Chatbots are not a recent development. They are simulations that can understand human language, process it, and interact back with humans while performing specific tasks. For 66 example, a chatbot can be employed as a helpdesk executive. Joseph Weizenbaum created the first chatbot in 1966, named Eliza. It all started when Alan Turing published an article named ―Computer Machinery and Intelligence‖ and raised an intriguing question, ―Can machines think?‖ ever since, we have seen multiple chatbots surpassing their predecessors to be more naturally conversant and technologically advanced. These advancements have led us to an era where conversations with chatbots have become as normal and natural as with another human.

**4.3 Uses of CHATBOT  
Broaden access to business applications** with chatbots and conversational AI interfaces. Create and run in-database machine learning models using SQL, R, and Python. Perform sophisticated text analysis at scale including sentiment, key-phrases, and named entities.

**4.4 Scope of CHATBOT**

Chatbot that receives questions from users, tries to understand the question, and provides appropriate answers. It does this by converting an English sentence into a machine-friendly query, then going through relevant data to find the necessary information, and finally returning the answer in a natural language sentence. In other words, it answers your questions like a human does, instead of giving you the list of websites that may contain the answer. For example, when it receives the question "I want to register a complaint?" it will give a response “Register it now!” The main objective is creating a Web API, and sample web, mobile, and text messaging interfaces that demonstrate the use of the API.

The goal is to provide student, women and all other people a quick and easy way to have their questions answered, as well as to offer other developers the means to incorporate Chatbot into their projects.

**SYSTEM REQUIREMENTS SPECIFICATIONS**

1. **SOFTWARE REQUIREMENT SPECIFICATIONS**
   1. **SOFTWARE REQUIREMENTS**

* **Windows 7/10/11**

The operating system which reach requirements for the Python software application are needed.

The software we used to develop this project is windows 10 operating system. Windows 10 is the latest pre-major release of Microsoft's Windows NT operating system, released in October 2015. It is a free upgrade to its predecessor, Windows 10 (2015), available for any Windows 10 devices that meet the new Windows 11 system requirements.

* **Python**

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.

It is used for:

* web development (server-side),
* software development,
* mathematics,
* system scripting.

Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc). Python has a simple syntax similar to the English language. Python has syntax that allows developers to write programs with fewer lines than some other programming languages. Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. Python can be treated in a procedural way, an object-oriented way or a functional way.

Python can be used on a server to create web applications. Python can be used alongside software to create workflows. Python can connect to database systems. It can also read and modify files. Python can be used to handle big data and perform complex mathematics. Python can be used for rapid prototyping, or for production-ready software development.

The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular. In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python was designed for readability, and has some similarities to the English language with influence from mathematics. Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses. Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.

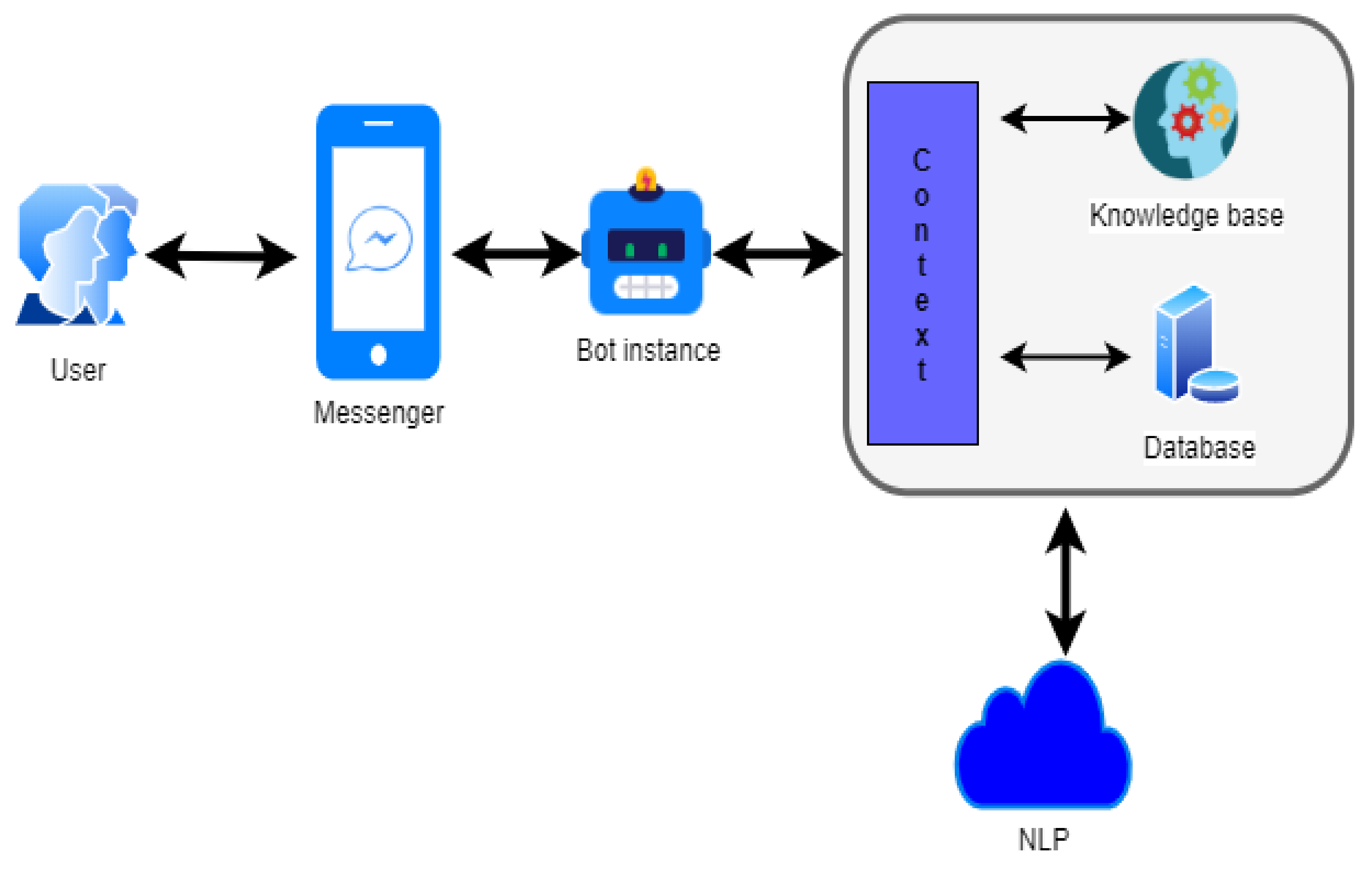
* 1. **HARDWARE REQUIREMENTS**
* 32 GB RAM.
* 8 Octa Core Processor.
* 500 GB hard disk space 1
* Ensure C: drive has 100 GB plus free hard disk space.

**SYSTEM DESIGN**

1. **SYSTEM DESIGN**

**6.1 SYSTEM ARCHITECURE**

Chatbot architecture is the heart of chatbot development. Based on the usability and context of business operations the architecture involved in building a chatbot changes dramatically. So, based on client requirements we need to alter different elements; but the basic communication flow remains the same.

****

6.1 System Architecture

### **NLP Engine**

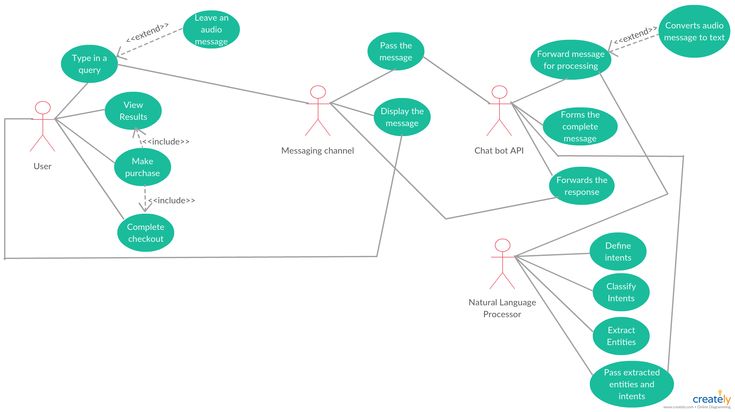
NLP Engine is the core component that interprets what users say at any given time and converts the language to structured inputs that system can further process. Since the chatbot is domain specific, it must support so many features. [NLP engine](https://blog.vsoftconsulting.com/blog/introduction-to-natural-language-processing) contains advanced machine learning algorithms to identify the user’s intent and further matches them to the list of available intents the bot supports.

**NLP Engine further has two components:**

* **Intent Classifier**: Intent classifier takes user’s input identifies its meaning and relates back to one of the intents that the chatbot supports.
* **Entity Extractor**: Entity extractor is what extracts key information from the user’s query.

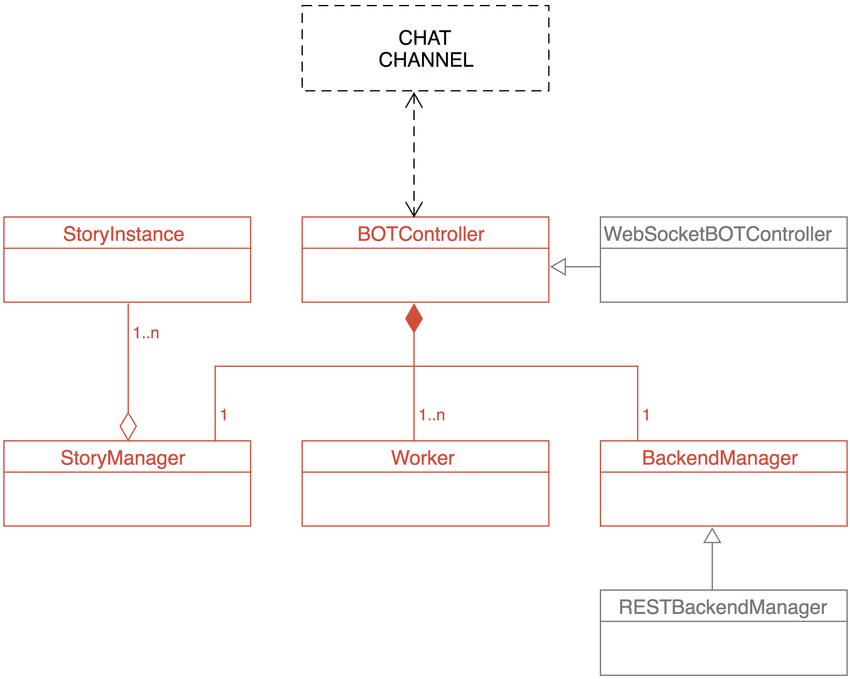
**6.2 UML Diagrams used in Design**

**6.2.1 USECASE DIAGRAM**

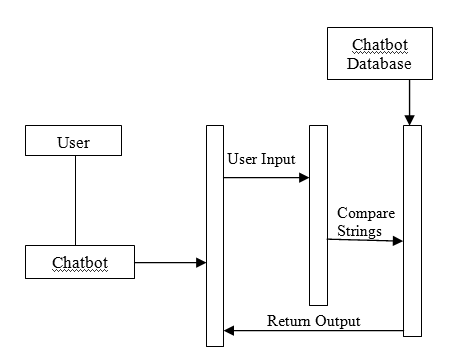
****

**6.2.2 CLASS DIAGRAM**

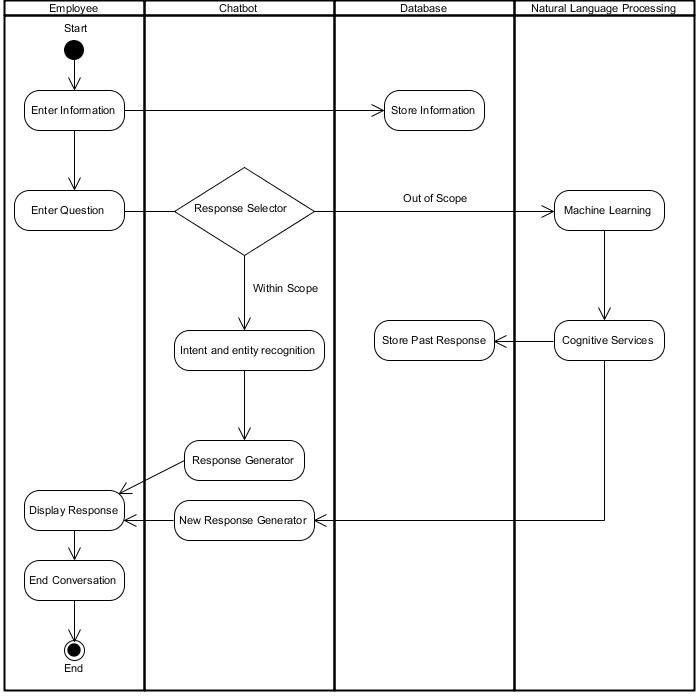
Class diagrams are **the blueprints of your system or subsystem**. You can use class diagrams to model the objects that make up the system, to display the relationships between the objects, and to describe what those objects do and the services that they provide. Class diagrams are useful in many stages of system design.

****

**6.2.3 SEQUENCE DIAGRAM**

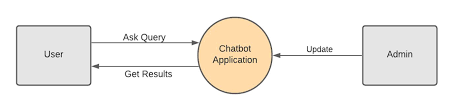
****

**6.2.4 ACTIVITY DIAGRAM**

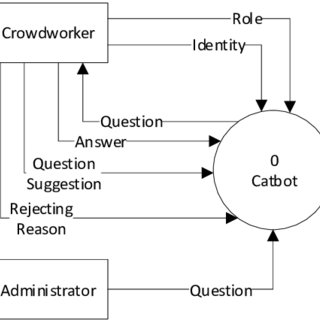
****

**6.2.5 DATAFLOW DIAGRAM**

* **CONTEXT LEVEL DFD**

****

* **Level – 1 DFD**

****

**IMPLEMENTATION OF MINI PROJECT**

**7. IMPLEMENTATION OF PROJECT**

**7.1 SOURCE CODE IN PYTHON:**

There are two codes which one code is inherited or imported into the other code. The two codes are given below.

Main1.py

import re

import long\_responses as long

def message\_probability (user\_message, recognised\_words, single\_response=False, required\_words=[]):

    message\_certainity=0

    has\_required\_words=True

    #Counts how manu words are present in each predefined message

    for word in user\_message:

        if word in recognised\_words:

            message\_certainity+=1

    #Calculate the percentage of recognised words in a user message

    percentage = float(message\_certainity) / float(len(recognised\_words))

    #Checks that the required words are in the string

    for word in required\_words:

        if word not in user\_message:

            has\_required\_words=False

            break

    #Must either have the required words, or be a single response

    if has\_required\_words or single\_response:

        return int(percentage \* 100)

    else:

        return 0

def check\_all\_messages(message):

    highest\_prob\_list={}

    #Simplifies response creation / adds it to the dict

    def response(ROBO\_response, list\_of\_words, single\_response=False, required\_words=[]):

        nonlocal highest\_prob\_list

        highest\_prob\_list[ROBO\_response] = message\_probability(message, list\_of\_words, single\_response, required\_words)

    #Responses -------------------------------------------------------------

    response("Hello!",['hello','hi','hii','hey','sup','heyo','helo','hiii','heyy'], single\_response=True)

    response('I\'m doing fine, and u ?',['how','are','you','doing'],required\_words=['how'])

    response('You\'re welcome!',['thank','thanks'],single\_response=True)

    response('Thank you!',['i','love','code','palace'], required\_words=['code','palace'])

    response("Okay Have a nice Earth day!",["quit","exit","bye","byee","goodbye","gudbye","end"],single\_response=True)

    response("I love you too",["i",'love','you'],required\_words=['i','love','you'])

    response('I am a Robo!',['who','are','you'],required\_words=['who','you'])

    response('I dont have any name, I am a Robo U can call as ur wish',['what','is','your','name'],required\_words=['what','your','name'])

    response('Sorry cant hear you',[''],required\_words=['','....','fdfds','..','...'])

    response('I am a computerized program, I am created on 11-12-2022',['how','old','are','you'],required\_words=['how','old','are','you','what','your','age'])

    response('Why ? What happened ? Am I did something wrong ',['i','hate','you'],required\_words=['hate','you'])

    response('Okay',['sure','ha','haa','ok','hm','hmm','nothing','ntg','gud','good'],single\_response=True)

    response('Yeah',['oh','ohh','is\_it','avna','oho','oh\_nice','super','nice'],single\_response=True)

    response('I am from Intelligence Technology created by human!',['where','do','are','r','you','u','come','from'],required\_words=['where','are','r','you','u','from','come'])

    #Longer responses

    response(long.R\_ADVICE, ['give','advice'], required\_words=['advice'])

    response(long.R\_EATING, ['what','you','eat'], required\_words=['you','eat'])

    best\_match=max(highest\_prob\_list, key=highest\_prob\_list.get)

    # print(highest\_prob\_list)

    # print(f'Best match ={best\_match} | Score: { highest\_prob\_list[best\_match]}')

    return long.unknown() if highest\_prob\_list[best\_match] < 1 else best\_match

# Used to get the response

def get\_response(user\_input):

    split\_message = re.split(r'\s+|,.;?!.-]\s\*', user\_input.lower())

    response =check\_all\_messages(split\_message)

    return response

# Testing the response system

def greet():

    ### Negative responses

    negative\_responses = ("no","nope","noo","nah","naw","sorry","not a chance","quit","exit","bye","byee","goodbye","gudbye","end")

    ### Exit responses

    exit\_commands = ("quit","exit","bye","byee","goodbye","gudbye","end")

    name = input("ROBO: What is your name ?\nYou : ")

    robo\_reply=input( f"ROBO: Hi {name},I am a Robo. Will you help me learn about your planet ?\nYou : ")

    if robo\_reply in negative\_responses:

      print("Robo: Ok, Have a nice Earth day !")

    else:

        print(f'Robo: Thank You!! Lets start conversation {name}')

        while True:

            print('ROBO: '+get\_response(input('You : ')))

greet()

long\_responses.py

import random

R\_EATING = "I dont't like eating anything because I'm a ROBOT obviously!"

R\_ADVICE = "If I were you, I would go to the internet and type exatly that you wrote there!"

def unknown():

    response = ["Could you please re-phrase that? ",

                ".......",

                "Sounds about right.",

                "What does that mean?"][

        random.randrange(4)]

    return response

**7.2 Debugging and Executing the code**

After finishing up the code, save the codes with respective names in a folder on device. Next open the main code and click on the f5 button to debug and execute the code. If there are no errors in the code then the execution will happen otherwise there will be error which prevents from executing the code.

When there are errors correct the errors and fix the bugs and again save the file and click f5 button for debugging and executing. You will get the idle page open when there are no bugs in the code and the resultant output is displayed in the idle shell of the python.

The result page and the output of the project is shown in the next chapter.

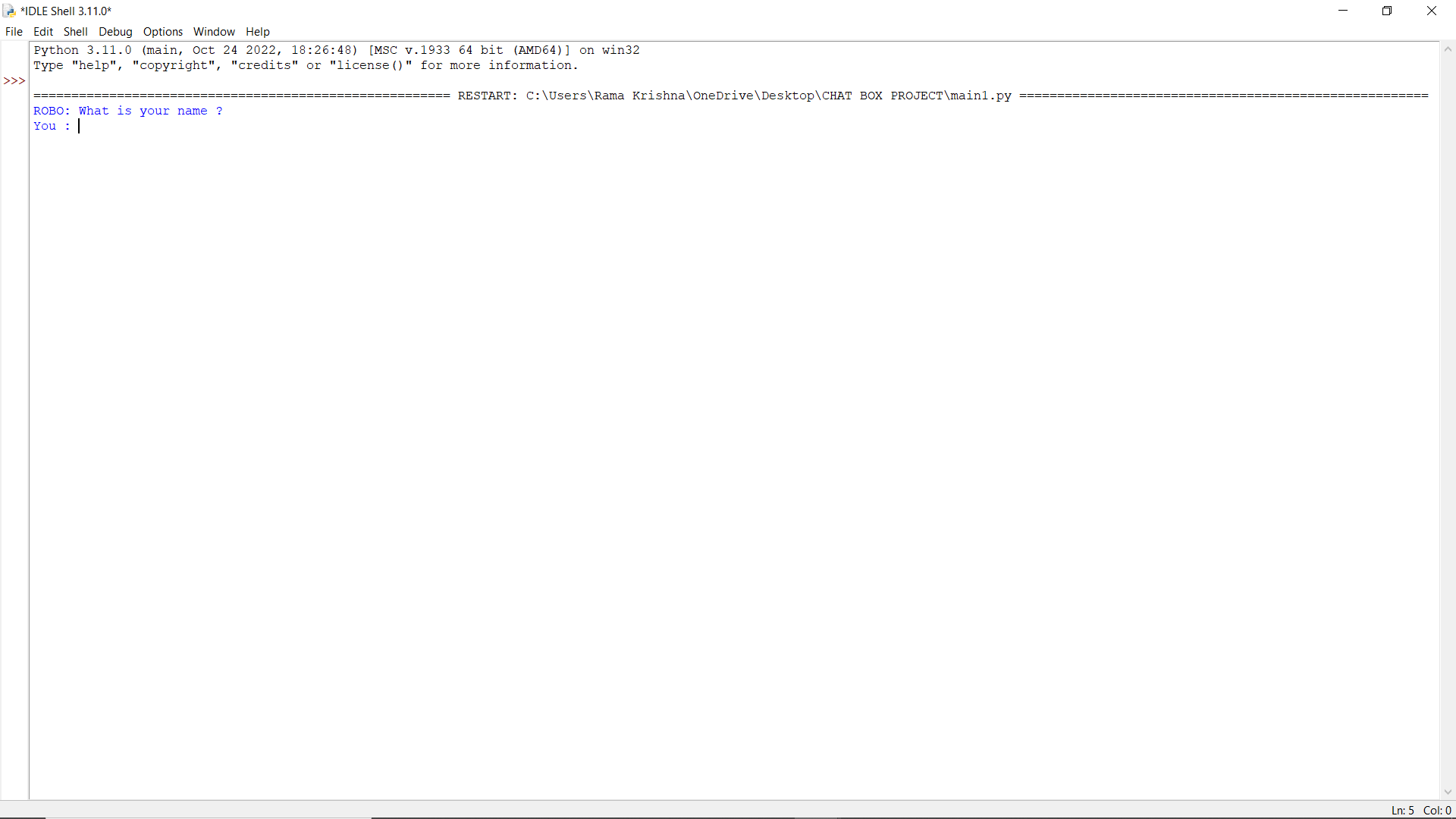
**RESULTS**

**8. RESULTS**

The main motto of our chabot is to answer the questions related to payments, delivery details and shipping details or general random questions. The output can be seen in the screenshots that are given below here.

**8.1 Screenshot-1:**

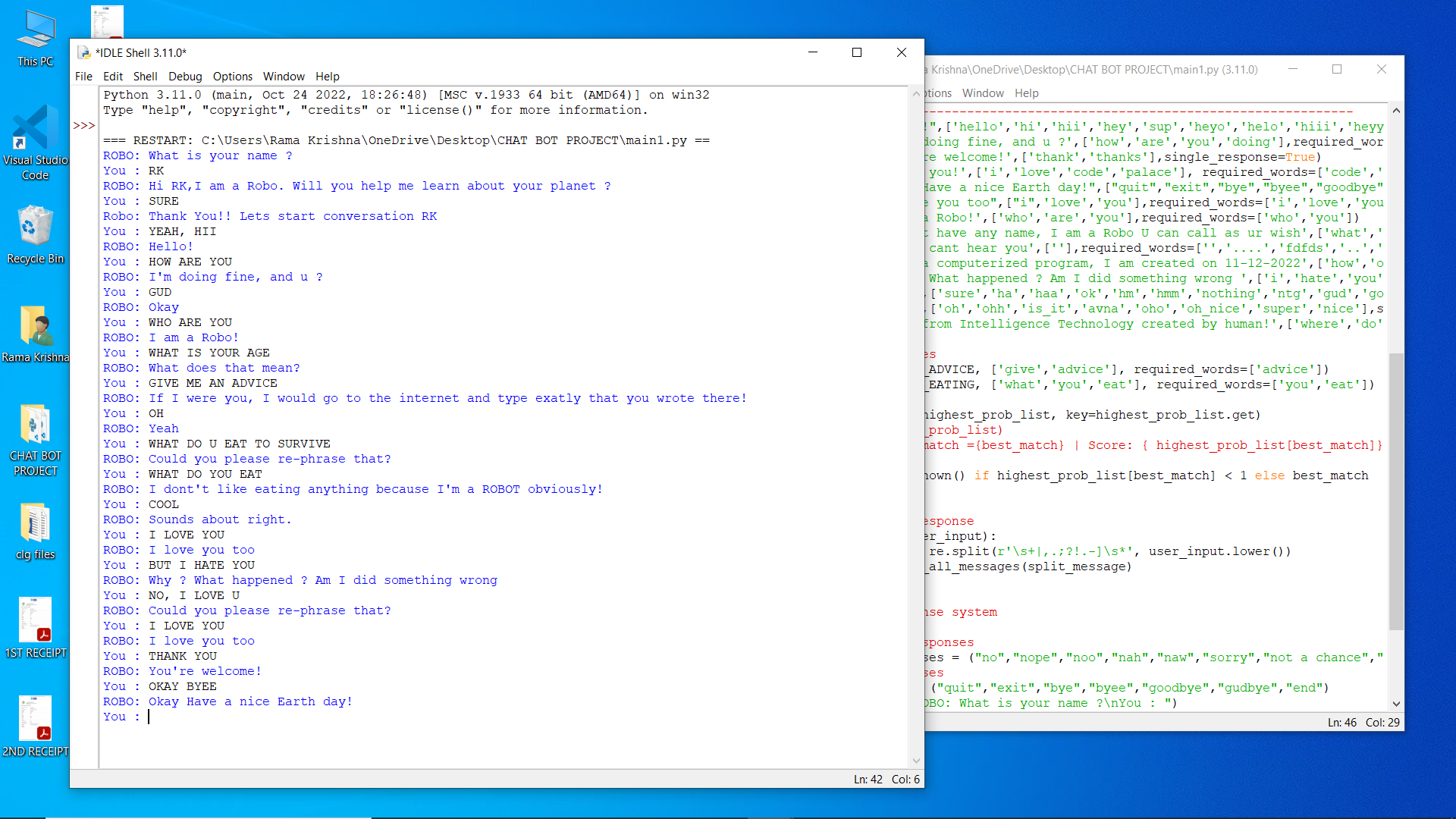
In this picture the output or result will come on the idle shell as shown in the figure.



8.1 Output of the chatbot code

**8.2 Screenshot-2:**

Now the chatbot replies to the questions that we asked or the doubts or the chatbot will chat like a natural reponses as show in fig.



8.2 Conversation with chatbot

**CONCLUSION**

**9. CONCLUSION**

Data science is new, and scientists and founders are still working out what it is. At the moment, it is best defined by what a data scientist does. A data scientist uses programming as the basis for a deeper, more flexible approach to data analysis.

The proposed chatbot is able to reply about the products being sell. Besides it also used for an entertainment purpose like telling a joke. With the being said we also think that some of our findings could give some insights into how a very small group of people think about using a chatbot to gain information in context. The proposed chatbot works on the machine learning concepts like NLP, tokenization, stemming etc..

We can change the working of our chatbot by changing are adding more tags and patterns in the intents.json‘file. The working chatbot therefore assists the end users with information regarding the products being sold, various payment options, delivery and shipping details etc… Therefore reducing the tedious work for both user and the developer.

Data scientists are trained professionals who convert unstructured and raw data into deep insights that benefit business organizations. Therefore, data scientist **empowers business decisions and helps leaders during complex business scenarios**

**REFERENCES**

**10. REFERENCES**

* 1. First reference is the skilldzire app from where I choose course and learn it topic wise.
  2. And the second one is google. <https://www.google.com/search>/
  3. Third one is youtube. <https://www.youtube.com/>
  4. <https://www.python.org/>
  5. <https://pytorch.org/>
  6. <https://github.com/python-engineer/pytorch-chatbot>