# Intelligent Personal Assistant

# J.A.R.V.I.S.

*Master Sambit Kumar Sahu*

*Adm. No.-202121107*

Department of CSA

Center For Post Graduate Studies

Odisha University of Agriculture and Technology

Bhubaneswar-2022

**Sahu S.K., 202121107, 2022, Intelligent Virtual Assistant-J.A.R.V.I.S.**

# Intelligent Personal Assistant

# J.A.R.V.I.S.

*A Project report submitted to the*
*Odisha University of Agriculture and Technology in partial fulfilment of the requirement for the*
*degree of Master of Computer Science and Application*
*By*

*(Master Sambit Kumar Sahu*
*Adm. No.: 202121107)*

Department of CSA
Center For Post Graduate Studies
Odisha University of Agriculture and Technology
Bhubaneswar-2022

Odisha University of Agriculture and Technology
Centre for Post Graduate Studies
Department of Computer Science and Application

# CERTIFICATE

This is to certify that the project report entitled **“Intelligent Virtual Assistant – J.A.R.V.I.S.”** submitted in partial fulfillment of the requirements for the award of the degree of Master of Computer Science and Application to the Odisha University of Agriculture and Technology is a faithful record of Bonafede and original research work carried out by **Master Sambit Kumar Sahu** under my guidance and supervision. No part of this Project has been submitted for any other degree or diploma. It is further certified that the assistance and help received by him from various sources during the course of investigation has been duly acknowledged.

**Dr. Rasmi Ranjan Patra**
Assistant Professor,
Department of Computer Science and Application

Odisha University of Agriculture and Technology
Centre for Post Graduate Studies
Department of Computer Science and Application

# CERTIFICATE

This is to certify that the project report entitled "**Intelligent Virtual Agent – J.A.R.V.I.S.**" submitted by **Master Sambit Kumar Sahu** to the Odisha University of Agriculture and Technology, Bhubaneswar in partial fulfillment of the requirements for the degree of Master of Computer Science has been approved/disapproved by the students' advisory committee and the external examiner.

## Advisory Committee

1. **Chairman:**

**Dr. Rasmi Ranjan Patra**
Assistant Professor,
Department of Computer Science and Application ……………………………………….

2. **Members:**

**Mr. Abhimanyu Dash**
Associate Professor,
Head-of-Department,
Department of Computer Science and Application ………………………………………..

**Dr. Sanjibani Sudha Pattanayak**
Assistant Professor,
Department of Computer Science and Application …………………………………………....

3. **External Examiner:**

………………………………………..

# ACKNOWLEDGEMENT

First and foremost, praises and thanks to the God, the Almighty, for his showers of blessings throughout my research work to complete the research successfully.

I am grateful to **Dr. Rasmi Ranjan Patra,** project guide, Centre for Post Graduate Studies, Odisha University of Agriculture and Technology, Bhubaneswar, for assigning me this innovation project and modelling me technically and morally for achieving greater success in life.

It is a great sense of satisfaction that my first real life venture in practical computing is in the form of project work. I extend my humble obligation towards **Mr. Abhimanyu Dash, Head of Department, Department of Computer Science and Application, Centre for Post Graduate Studies, Odisha University of Agriculture and Technology,** for providing us with an environment to study and build our career.

I express my profound gratitude and indebtedness to **Dr. Sanjibani Sudha Pattanayak, Assistant Professor, Department of Computer Science and Application,** for her constant support and motivation, **Dr. Jyoti Ranjan Mohanty, Assistant Professor, Department of Computer Science and Application,** for his useful suggestions given during the course of project period, **Mrs. Sarita Mohanty, Assistant Professor, Department of Computer Science and Application,** for her unending support right from the conceptualization of the idea till the evolution of the software, Mrs. **Tanuja Panda, Assistant Professor, Department of Computer Science and Application,** for her understanding and assistance, **Dr. Subhadra Mishra, Assistant Professor, Department of Computer Science and Application,** for her vital encouragement and support. Last but not in the least **Mrs. Debaswapna Mishra, Assistant Professor, Department of Computer Science and Application,** for the help and inspiration he extended.

Finally, my thanks go to all the people who have supported me to complete the research work directly or indirectly.

**Sambit Kumar Sahu**
Regn. No. :- 202121107

# DECLARATION

I, Sambit Kumar Sahu, a student of Master’s in Computer Science and Application, Centre for Post Graduate Studies, Odisha University of Agriculture and Technology, Bhubaneswar hereby declare that the project report entitled “Intelligent Virtual Assistant – J.A.R.V.I.S.” has been done by me under the guidance of Dr. Rasmi Ranjan Patra, Project Guide, Centre for Post Graduate Studies, Odisha University of Agriculture and Technology, Bhubaneswar for the partial requirement for the award of the degree of Master’s in Computer Science and Application.

This Project submitted by me is the effort of mine and has not been deposited to any other organization or published earlier.

**Place:- Bhubaneswar**
**Date:- 16/12/2022**

**Sambit Kumar Sahu**
**Regn. No.: - 202121107**

# ABSTRACT

As we know Python is an emerging language, so it becomes easy to write a script for Voice Assistant in Python. The instructions for the assistant can be handled as per the requirement of the user. Speech recognition is the process of converting speech into text. This is commonly used in voice assistants like Alexa, Siri, etc. In Python there is an API called SpeechRecognition which allows us to convert speech into text. It was an interesting task to make my own assistant. It became easier to search on Google without opening the browser, and performing many other daily tasks like playing music, opening your favorite IDE with the help of a single voice command. In the current scenario, advancement in technologies are such that they can perform any task with same effectiveness or can say more effectively than us. By making this project, I realized that the concept of AI in every field is decreasing human effort and saving time.

The Functionalities of this project includes provides current news, talk about your current location, make a phone call, can open your favorite Integrated Development Environment, Notepad++, etc., Hide/Unhide files, provide System Details and IP address, etc. , can do Wikipedia searches for you, can open websites like Google, YouTube, etc., in a web browser, can give weather forecasts., volume adjustment, take and delete screenshots and many more, find places, switch windows, etc.

The user statements/commands are analyzed with the help of machine learning to give an optimal solution. The main purpose of A.I machines is that it can perform human tasks with the same efficiency or even more efficiently than humans. It is a fact that my virtual assistant is a particularly good example of artificial intelligence.

# TABLE OF CONTENTS

Page No.

# CHAPTER-1

## (INTRODUCTION)

# INTRODUCTION

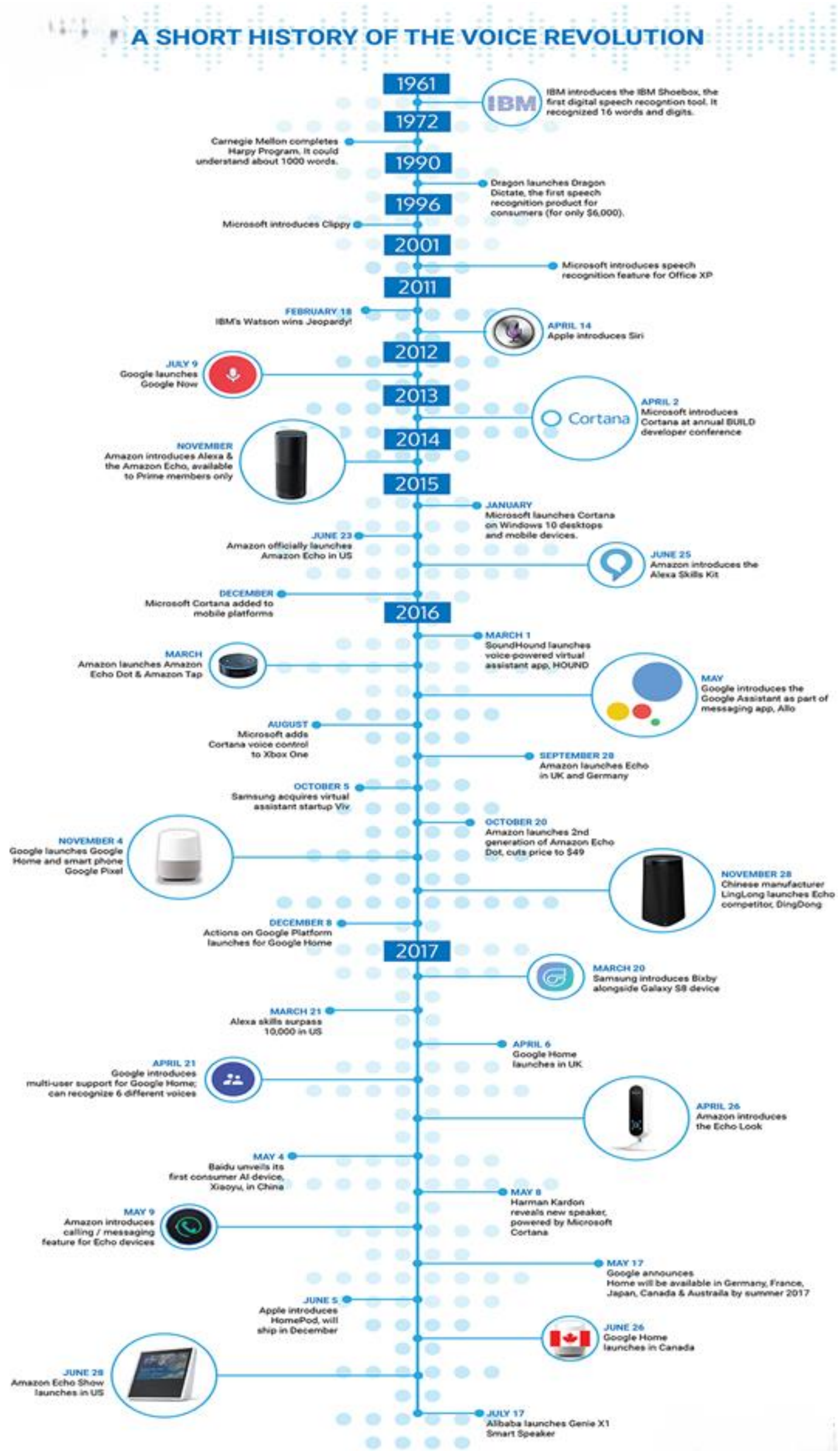
**Intelligent virtual agents** are **AI-powered software that can engage with customers in a conversational way**. They are digital assistants that can understand human speech and respond the same way a person would, especially when it comes to simple, repetitive requests. Although sometimes confused with chatbots, **virtual agents are actually much more sophisticated**. And since recent advancements in AI are making IVAs easy-to-use and cost-effective, more and more businesses are adopting virtual agents instead of simple chatbots. Chatbots imitate human dialogue to a certain extent but are limited by a given script. Whereas IVA's initiate context-related, human-like dialogues.

By using machine learning and deep learning, virtual agents are able to build up a large vocabulary, understand slang and misspelled words, and learn from each experience. This means that they are able to give **precise and personalized responses** to inquiries and only get better as time goes on. That is a clear advantage compared to traditional chatbots, which can only provide scripted responses and are unable to learn and grow. The conversational skills are powered by cognitive process automation and natural language understanding, a subset of artificial intelligence that mimic the way the human brain works - to assist humans in making decisions, completing tasks, or meeting goals.

Cognitive Process Automation uses technologies such as natural language processing (NLU/NLG), image processing, pattern recognition, and - most importantly - contextual analysis to make more intuitive leaps, perceptions, and judgments for better speech recognition. In short, an Intelligent Virtual Agent is not a basic bot that will only feed the user a limited set of preregistered answers.

A Virtual Agent's conversational skills enables better language understanding of the user, ask additional questions to understand a context (who the user is, his role in the organization) and orchestrate multiple actions.

As the voice assistant is using Artificial Intelligence, the results that it is providing are highly accurate and efficient. The assistant can help to reduce human effort and consumes time while performing any task, they removed the concept of typing completely and behave as another individual to whom we are talking and asking to perform task. The assistant is no less than a human assistant, but we can say that this is more effective and efficient to perform any task. The libraries and packages used to make this assistant focus on the time complexities and reduces time.



A SHORT HISTORY OF THE VOICE REVOLUTION

1961

IBM introduces the IBM Shoebox, the first digital speech recogntion tool. It recognized 16 words and digits.

1972

Carnegie Mellon completes Harpy Program. It could understand about 1000 words.

1990

Dragon launches Dragon Dictate, the first speech recognition product for consumers (for only $6,000).

1996

Microsoft introduces Clippy

2001

Microsoft introduces speech recognition feature for Office XP

2011

FEBRUARY 18
IBM's Watson wins Jeopardy!

APRIL 14
Apple introduces Siri

2012

JULY 9
Google launches Google Now

2013

APRIL 2
Microsoft introduces Cortana at annual BUILD developer conference

2014

NOVEMBER
Amazon introduces Alexa & the Amazon Echo, available to Prime members only

2015

JANUARY
Microsoft launches Cortana on Windows 10 desktops and mobile devices.

JUNE 23
Amazon officially launches Amazon Echo in US

JUNE 25
Amazon introduces the Alexa Skills Kit

DECEMBER
Microsoft Cortana added to mobile platforms

2016

MARCH 1
SoundHound launches voice-powered virtual assistant app, HOUND

MARCH
Amazon launches Amazon Echo Dot & Amazon Tap

MAY
Google introduces the Google Assistant as part of messaging app, Allo

AUGUST
Microsoft adds Cortana voice control to Xbox One

SEPTEMBER 28
Amazon launches Echo in UK and Germany

OCTOBER 5
Samsung acquires virtual assistant startup Viv

OCTOBER 20
Amazon launches 2nd generation of Amazon Echo Dot, cuts price to $49

NOVEMBER 4
Google launches Google Home and smart phone Google Pixel

NOVEMBER 28
Chinese manufacturer LingLong launches Echo competitor, DingDong

DECEMBER 8
Actions on Google Platform launches for Google Home

2017

MARCH 20
Samsung introduces Bixby alongside Galaxy S8 device

MARCH 21
Alexa skills surpass 10,000 in US

APRIL 6
Google Home launches in UK

APRIL 21
Google introduces multi-user support for Google Home; can recognize 6 different voices

APRIL 26
Amazon introduces the Echo Look

MAY 4
Baidu unveils its first consumer AI device, Xiaoyu, in China

MAY 8
Harman Kardon reveals new speaker, powered by Microsoft Cortana

MAY 9
Amazon introduces calling / messaging feature for Echo devices

MAY 17
Google announces Home will be available in Germany, France, Japan, Canada & Austraila by summer 2017

JUNE 5
Apple introduces HomePod, will ship in December

JUNE 26
Google Home launches in Canada

JUNE 28
Amazon Echo Show launches in US

JULY 17
Alibaba launches Genie X1 Smart Speaker

A figure from www.voicebot.ai

## 1.1. EVOLUTION

In the early stages of chatbot development, the core NLP method was used to design the chatbots as Machine Learning was not exactly feasible. By the time the machine learning methods came into effect and use which channels more data and code. Let us look at the timeline from 1995 – where an NLP based bot was launched, which is the first bot with the knowledge of pattern was matching rules to human input.

- **1995 (ALICE):** An NLP bot which applied heuristic pattern matching rules to human input called ALICE – Artificial Linguistic Internet Computer Entity
- **2000 (Smarterchild):** a chatbot – which was available on AOL Instant Messenger and Windows Live Messenger networks was developed by ActiveBuddy. This bot is a precursor to Apple's Siri and Samsung's S Voice in many ways.
- **2011 (Siri by Apple):** a voice-activated intelligent assistant was launched by Apple as part of its iOS and macOS platforms.
- **2012 (Google Now):** Google develops chatbot for Google searches mobile app named as Google Now.
- **2014 (Alexa by Amazon, Cortana by Microsoft):** Amazon launched Alexa, an intelligent personal assistant via the Amazon Echo. And Microsoft launches Cortana, a virtual assistant named after the fictional character from "Halo".
- **2015 (M by Facebook):** Facebook launched a hybrid bot-and-human virtual assistant called M which is accessible through Messenger.
- **2016 (Google Home, Sirikit by Apple):** Google unveiled its answer to Amazon Echo called Google Home. Apple introduced Sirikit.
- **2017 (Google Assistant, Woebot by Woebot Labs)**: it expands beyond Android to iOS and is available now for download on the iTunes stores for iPhones and Google Home apart from scheduling appointments it can make hands-free calls as well.

# CHAPTER-2

## (LITERATURE STUDY)

# LITERATURE STUDY

## 2.1. A Study into Preferred Explanations of Virtual Agent Behavior.

Virtual training systems provide an effective means to train people for complex, dynamic tasks such as crisis management or firefighting. Intelligent agents are often used to play the characters with whom a trainee interacts. To increase the trainee's understanding of played scenarios, several accounts of agents that can explain the reasons for their actions have been proposed. This paper describes an empirical study of what instructors consider useful agent explanations for trainees. It was found that different explanation types were preferred for different actions, e.g., conditions enabling action execution, goals underlying an action, or goals that become achievable after action execution. When an action has important consequences for other agents, instructors suggest that the others' perspectives should be part of the explanation.

## 2.2. A Study on Speech Recognition System.

To be able to control devices by voice has always intrigued mankind. Today after intense research, Speech Recognition System, have made a niche for themselves and can be seen in many walks of life. The accuracy of Speech Recognition Systems remains one of the most important research challenges e.g. noise, speaker variability, language variability, vocabulary size and domain. The design of speech recognition system requires careful attentions to the challenges such as various types of Speech Classes and Speech Representation, Speech Preprocessing stages, Feature Extraction techniques, Database and Performance evaluation.

## 2.3. Natural Language Processing: State of The Art, Current Trends and Challenges.

Natural language processing (NLP) has recently gained much attention for representing and analyzing human language computationally. It has spread its applications in various fields such as machine translation, email spam detection, information extraction, summarization, medical, and question answering etc. The paper distinguishes four phases by discussing different levels of NLP and components of Natural Language Generation (NLG) followed by presenting the history and evolution of NLP, state of the art presenting the various applications of NLP and current trends and challenges.

## 2.4. Overview of Neural Networks.

Artificial Neural Networks (ANN) is inspired by the human brain and it's can be used for machine learning and artificial intelligence. With these networks, various problems can be solved computer based. The artificial neural network (ANN) is to some extent modelled on the structure of the biological brain. It consists of an abstracted model of interconnected neurons, whose special arrangement and linking can be used to solve computer-based application problems in various fields such as statistics, technology or economics. The neural network is a research subject of Neuro informatics and part of the artificial intelligence. Neural networks must be trained before they can solve problems.

### 2.5. A Study on Cross Entropy Error Function in Neural Networks.

The ANN is implemented using the cross entropy error function in the training stage. The cross entropy function is proven to accelerate the backpropagation algorithm and to provide good overall network performance with relatively short stagnation periods. Forecasting performance measures such as mean square errors (MSE) and mean absolute deviations (MAD) are presented for models.

### 2.6. Comparative Analysis of Optimizers in Deep Neural Networks

The role of optimizer in deep neural networks model impacts the accuracy of the model. Deep learning comes under the umbrella of parametric approaches; however, it tries to relax as many as assumptions as possible. The process of obtaining parameters from the data is gradient descent. Gradient descent is the chosen optimizer in neural network and many of the machine learning algorithms. The classical stochastic gradient descent (SGD) and SGD with momentum which were used in deep neural networks had several challenges which were attempted to resolve using adaptive learning optimizers. Adaptive learning algorithms like- RMSprop, Adagrad, Adam wherein learning rate for each parameter is computed were further developments for better optimizer. Adam optimizer in Deep Neural Networks is often a default choice observed recently. Adam optimizer is a combination of RMSprop and momentum. Though, Adam since its introduction has gained popularity, there are claims that report convergence problem with Adam optimizer. Also, it is advocated that SGD with momentum gives better performance compared to Adam. This paper presents comparative analysis of SGD, SGD with momentum, RMSprop, Adagrad and Adam optimizer on weather dataset. The dataset was processed assuming Adam optimizer will prove to be the better optimizer choice as preferred a default choice by many, however, SGD with momentum proved to be a unsurpassed optimizer for this particular dataset.

### 2.7. Comparison of concrete strength prediction techniques with artificial neural network approach.

Prediction of concrete strength is an important issue in ready-mixed concrete industry, especially in proportioning new mixtures and for the quality assurance of the concrete produced. In this paper, it is aimed to illustrate that the artificial neural networks can be used for predicting the 28-day strength of low to medium strength concretes. The compositional, fresh concrete and early strength data obtained from different batching plants of a ready-mixed concrete company have been defined in terms of ten independent variables that are grouped in five different system models to which neural network and multiple linear regression models have been applied. The accuracies of prediction by artificial neural network and multiple linear regression models as well as by Abrams' law are compared on the basis of the coefficient of determination. It appears that the best results are obtained by the artificial neural network models using data for fresh concrete and early strength simultaneously.

### 2.8. A Study on Qt Designer IDE.

Among the many GUI programming libraries available, both free and licensed, the Qt C++ Library from Trolltech has a strong following. And with the advent of Qt designer 3.3-1, Qt developers now have a feature-rich IDE with which to design and code GUI applications. Designer works with Qt project files (.pro) directly so that it can be used on an as-needed basis to design and implement the GUI portions of your application. Designer's project management gives you control over application include paths, linking with other libraries, and platform-specific settings.

### 2.9. A Study on NumPy

NumPy is a Python library optimized for numerical computing. It bears close semblance with MATLAB and is equally as powerful when used in conjunction with other packages such as SciPy for various scientific functions, Matplotlib for visualization, and Pandas for data analysis. NumPy is short for numerical python.

### 2.10 Study on PyTorch

PyTorch is a library for Python programs that encourages deep learning programs. With this receptiveness and convenience found in (Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras. PyTorch makes it useful in developing deep neural networks. It has an expansive scope and is applied for various applications. As Python is for programming, PyTorch is a magnificent prologue to profound learning just as an instrument usable in proficient real-world applications.

### 2.11. A Study on Application Programming Interfaces(APIs).

Application Programming Interface (API) is packed functionality to solve specific tasks. In order for developers to learn to use its functionality, APIs include some kind of documentation. Documentation is an important part of the API itself, but providing high-quality documentation is not a straightforward task. Nowadays, most of the documentation does not include the information expected by users. Another problem is the lack of comprehensive evaluation methods that can help creators to identify missing or incomplete elements in their documentation.

### 2.12. A Comparative Study on Python Modules and Packages.

Modules and packages are two constructs used in Python to organize larger programs. This chapter introduces modules in Python, how they are accessed, how they are defined and how Python finds modules etc. It also explores Python packages and sub-packages.

# CHAPTER-3

## (MOTIVATION)

# MOTIVATION

It is named the “Job of the Century” It helps you earn dollars. It opens a world of opportunities. Absolutely! You agree and you are aware of all these advantages of learning Artificial Intelligence. But AI is beyond and ahead in every niche, it has stepped in. We are in an age where call Alexa/Google/Siri more than we call out our friends or family. Artificial Intelligence is no longer limited to science-fiction movies or shows. It is all around.

It is the Gmail reply prompter, Amazon purchase recommendations, Spotify music suggestions, auto appointment booking for your next haircut, and of course, our precious AI assistants. That is the thing about AI. It analyses your earlier playlist at different timelines, your browsing history, previous orders, the places you have been, and much more to derive behavioral models. If you are someone who wants to learn the nit and grit of Artificial Intelligence, let us tell you. It is not going to be a “cakewalk” but if you are passionate and consistent, the future welcomes you!

## 3.1. The Skill of the Century

Not year, not decade, a century! It is often heard that AI is all set to replace a lot of jobs that humans do. True! But it is also creating more than 130 million roles in all major sectors. So, in order to be a part of the group, you need to be a part of this transformation. Artificial Intelligence is one of the emerging technologies making its mark in every industry ranging from fashion to finance.

## 3.2. AI is Everywhere

An automated computer program that engages/interacts with the website visitors like a human and costs one little to nothing. The major reason the organization is turning towards chatbots is that they are active and live 24*7 across the globe. Any chatbot responses would be based on the information fed to it and the content available on the website.

If you get updated with the contemporary tech news, odds are you identify the concept of self-driving cars. Ever been in a breathtaking situation when the driver lost a little bit of control of the vehicle? Imagine what it feels like to travel without a driver at all. Scary? Not anymore. The Automobile Industry has already designed such cars that are enjoying test rides. But the talk of the town is “self-driving” are going to hit the roads sooner than we expected.

Smart home devices are now changing our home environment based on our preferences. These machine learning and artificial intelligence build devices can change the temperature, dim or brighten lights, maximize home security, etcetera.

The Automobile Industry, Music Recommendations, Smart Home Devices, Online Customer Support, Security Surveillance, Retail, and Healthcare are just a few industries to name. Yes, it is everywhere.

### 3.3. Data, Data, More Data

We, humans, generate more than 2.5 quintillion bytes every single freaking day. Can you believe that? We collect from what we consume when we see, how we communicate. We feed the collected data to machine learning algorithms to retrieve a behavioral pattern also known as some sort of information about consumers. Information that could transform as insights. Therefore, companies are literally running towards AI in the hope to make more sales and win the race ultimately.

### 3.4. AI is versatile

As cliche as it sounds, artificial intelligence has gotten everything in the bag to stand out from the crowd. The mere mention of artificial intelligence draws our minds to either computer or space-related industries. But AI also has a major role in industries like banking, healthcare, security, mobile, fraud detection, clothing, and much more.

The above advantages encourages me to work in this Artificial Intelligence Project – J.A.R.V.I.S..

# CHAPTER-4

## (OBJECTIVES)

# OBJECTIVES

The project aims to develop a personal assistant for desktop. JARVIS draws its inspiration from virtual assistants like google assistants like google assistant for android, and Siri for iOS. It has been designed to provide a user-friendly interface for carrying out a variety of tasks by employing certain well-defined commands. Users can interact with the assistant through voice commands. As a personal assistant, JARVIS assists the end-user with day to day activities like general human conversation, searching queries in google, launch applications, live weather conditions, search places and calculate from current location, take screenshots and many more.

- This Software aims at developing a personal assistant for windows.
- The main purpose of the software is to perform the tasks of the user at certain commands, provided via speech.
- It will ease most of the work of the user as a complete task can be done on a single command.
- JARVIS draws its inspiration from Virtual Assistants and Siri for iOS.
- Users can interact with the assistant through voice commands.

J.A.R.V.I.S. is acronym of Just a Rather Very Intelligent System. J.A.R.V.I.S. is an artificial intelligence created by Tony Stark, who later controls his Iron Man and Hulk buster armor for him. It is an Intelligent Virtual Assistant that allows customers to quickly find relevant information or complete tasks based on individual information. These platforms utilize natural language processing, also known as NLP, which helps computers understand text and spoken words similar to humans.

# CHAPTER-5

## (TOOLS AND TECHNOLOGIES)

# TOOLS AND TECHNOLOGIES

The IDE used in this project is Jupyter. All the python files were created in Jupyter and all the necessary packages were easily installable in this IDE. For this project the following modules and libraries were used i.e. pyttsx3, SpeechRecognition, datetime, wikipedia, pywhatkit, pyjokes, pyautogui, pyQt, torch, json, signal, nltk, numpy, subprocess, twillio, geopy, etc. I have created a live GUI for interacting with the JARVIS as it gives a design and interesting look while having the conversation.

1. **Jupyter** :- Jupyter notebook is an open-source IDE that is used to create Jupyter documents that can be created and shared with live codes. Also, it is a web-based interactive computational environment. The Jupyter notebook can support various languages that are popular in data science such as Python, Julia, Scala, R, etc.

2. **PyQT5(Live GUI)** :- PyQt5 is the most important python binding. It contains a set of GUI widgets. PyQt5 has some important python modules like QtWidgets,QtCore,QtGui, QtDesigner,etc.

3. **Python Libraries** :- In J.A.R.V.I.S. following python libraries were used:

    - pyttsx3:- It is a python library which converts text to speech.

    - speech_recognition :- It is a python module which converts speech to text.

    - pywhatkit :- It is most popular python library for YouTube automation. It’s easy to use and does not require you to any additional setup.

    - datetime :- This library provides us with the actual date and time.

    - wikipedia :- It is a python module for searching anything in wikipedia,

    - asyncio :- asyncio is a library to write concurrent code using the async/await syntax. asyncio is used as a foundation for multiple Python asynchronous frameworks that provide high-performance network and web-servers, database connection libraries, distributed task queues, etc.

- socket :- Sockets and the socket API are used to send messages across a network. They provide a form of inter-process communication (IPC). The network can be a logical, local network to the computer, or one that's physically connected to an external network, with its own connections to other networks.

- pprint :- The pprint module provides a capability to “pretty-print” arbitrary Python data structures in a form which can be used as input to the interpreter. If the formatted structures include objects which are not fundamental Python types, the representation may not be loadable. This may be the case if objects such as files, sockets or classes are included, as well as many other objects which are not representable as Python literals.

- pyautogui :- PyAutoGUI is essentially a Python package that works across Windows, MacOS X and Linux which provides the ability to simulate mouse cursor moves and clicks as well as keyboard button presses.

- json :- Python has a built-in package called json, which can be used to work with JSON data. JSON is a syntax for storing and exchanging data. JSON is text, written with JavaScript object notation.

- geopy :- geopy is a Python client for several popular geocoding web services. Geopy makes it easy for Python developers to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources. Geopy includes geocoder classes for the OpenStreetMap Nominatim, Google Geocoding API (V3), and many other geocoding services. The full list is available on the Geocoders doc section. Geocoder classes are located in geopy.geocoders.

- time :- As the name suggests Python time module allows to work with time in Python. It allows functionality like getting the current time, pausing the Program from executing, etc.

- PIL :- The Python Imaging Library adds image processing capabilities to your Python interpreter. This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities. The core image library is designed for fast access to data stored in a few basic pixel formats. It should provide a solid foundation for a general image processing tool.

- os :- It represents Operating System related functionality.

- sys :- It allows operating on the interpreter as it provides access to the variables and functions that usually interact strongly with the interpreter.

- torch :- The torch package contains data structures for multi-dimensional tensors and defines mathematical operations over these tensors. Additionally, it provides many utilities for efficient serializing of Tensors and arbitrary types, and other useful utilities.

4. **Functions:-** The member functions that are used for specific problem.

- Listen() :- The function is used to take the command as input through microphone of user and returns the output as string.

- wish() :- This function greets the user according to the time like Good Morning, Good Afternoon and Good Evening.

- NonInputExecution(query) :- This function provides single value as current time, current date and current day.

- InputExecution(tag,query) :- This is the function which contains all the necessary task execution definition like launch_apps(), website_opener(), news(), my_location(), weather() and many conditions in if condition like “open google”, “open notepad”, “search on wikipedia” ,”get weather” and “make a phone call” etc.

- Tokenize(sentence) :- It actually returns the syllables from a single word. A single word can contain one or two syllables.

- stem(word):- stemming is the process of producing morphological variants of a root/base word. Stemming programs are commonly referred to as stemming algorithms or stemmers. A stemming algorithm reduces the words “chocolates”, “chocolatey”, and “choco” to the root word, “chocolate” and “retrieval”, “retrieved”, “retrieves” reduce to the stem “retrieve”.

- bag_of_words(tokenized_sentence,words) :- Bag of Words model is used to preprocess the text by converting it into a bag of words, which keeps a count of the total occurrences of most frequently used words.

5. **Application Programming Interface** :- API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. In J.A.R.V.I.S., following API calls are used, having of specific keys:-

    - newsapi :- News API is a simple, easy-to-use REST API that returns JSON search results for current and historic news articles published by over 80,000 worldwide sources.

    - Openweathermapapi:- It is an Application Programming Interface that allows weather data to be queried from scripts and code**.**

    - twillioapi :- Twilio's APIs (Application Programming Interfaces) power its platform for communications. Behind these APIs is a software layer connecting and optimizing communications networks around the world to allow your users to call and message anyone, globally. Twilio has a whole host of APIs, from SMS to Voice to Wireless.

# CHAPTER-6

## (SYSTEM DESIGN)

## MODEL OF INTELLIGENT VIRTUAL ASSISTANT

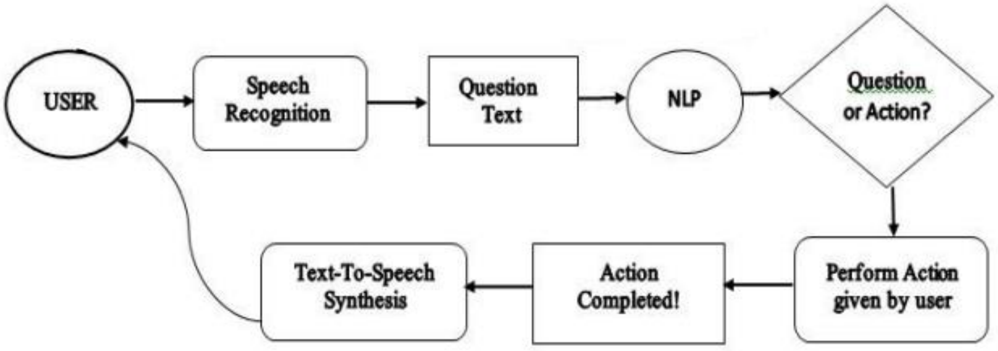


Fig 6.2. A figure on data flow.

The model consists of user input through microphone to accept commands from the user. These commands are then go through Speech Recognition, it is the ability of a machine or program to identify words and phrases in spoken languages and convert them to a machine-readable format. On these input Natural Language Processing is applied, it is a field which is created by amalgamating computer science and artificial intelligence. Using NLP, we are concerned with interactions between computers and human natural languages. Our Voice assistant uses text-to-speech to understand all the words spoken by the user, and based on certain conditions that satisfy being a command the voice assistant sends responses to the user.

The user gives the input in the form of voice; this voice command is recognized by the application. Then it will check whether it is the authorized user, then the action is performed as per the command given by the user. Command given is compared as a form of action and question and respond with the dialog box or search through the knowledge base.

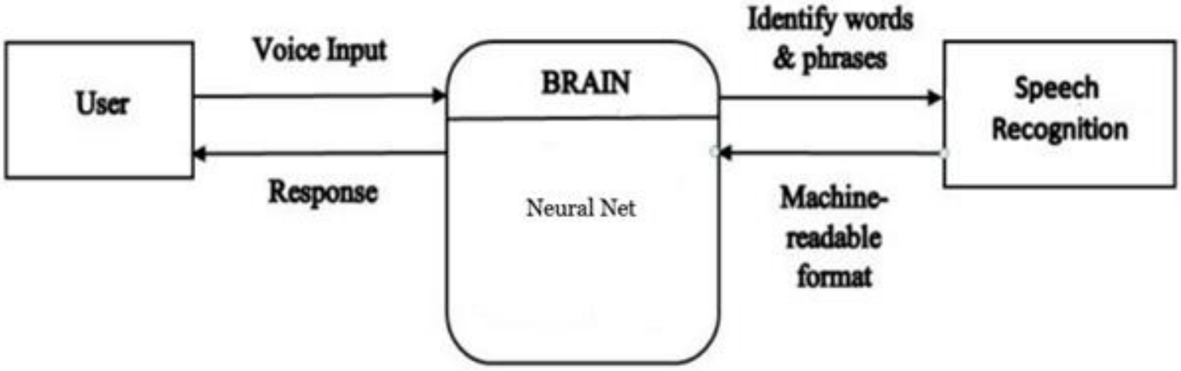


Fig 6.2. A figure on data flow.

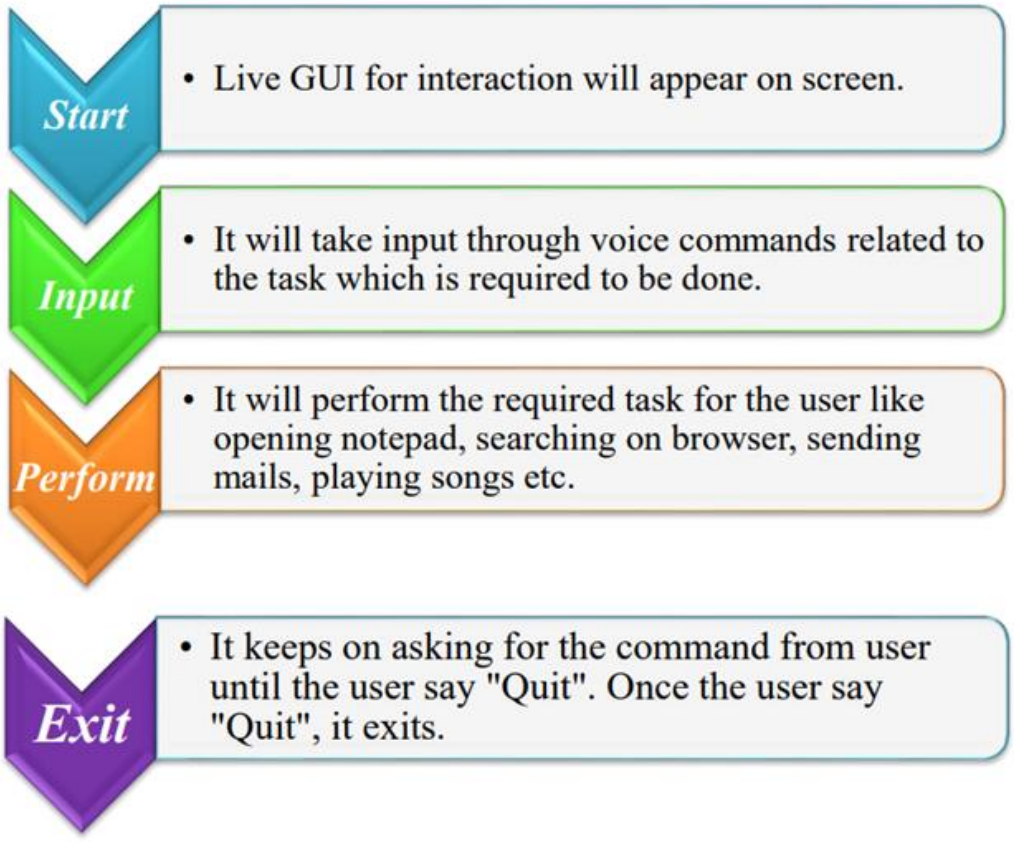
Input is given by user in the form of voice. SpeechRecognition will convert this voice data into text form and then the action is performed by the voice assistant according to the command given by the user by comparing with the dialog box and knowledge base.

# CHAPTER-7

## (WORKFLOW & METHODOLOGIES)

# WORKFLOW

The work flow for JARVIS is as follow:



- Live GUI for interaction will appear on screen.
- It will take input through voice commands related to the task which is required to be done.
- It will perform the required task for the user like opening notepad, searching on browser, sending mails, playing songs etc.
- It keeps on asking for the command from user until the user say "Quit". Once the user say "Quit", it exits.

The system is designed using the concept of Artificial Intelligence and with the help of necessary packages of Python. Python provides many libraries and packages to perform the tasks, for example pywhatkit is basically used for Youtube video handling, etc.

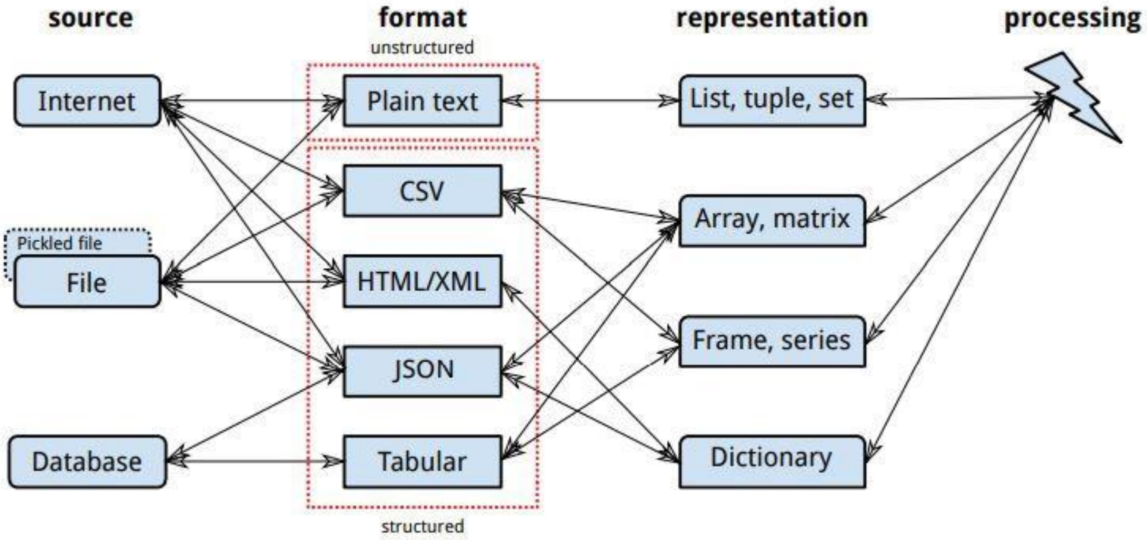
The data in this project is nothing but user input, whatever the user says, the assistant performs the task accordingly. The user input is nothing specific but the list of tasks which a user wants to get performed in human language i.e. English.

## 1. Data Acquisition :-

Data acquisition is all about obtaining the artifacts that contain the input data from a variety of sources, extracting the data from the artifacts, and converting it into representations suitable for further processing.

The three main sources of data are the Internet (namely, the World Wide Web), databases, and local files (possibly previously downloaded by hand or using additional software).

- Unstructured plain text in a natural language (such as English or Chinese)
- Structured data, including Tabular data in comma separated values (CSV) files, Tabular data from databases, tagged data in HyperText Markup Language (HTML) or, in general, in eXtensible Markup Language (XML), Tagged data in JavaScript Object Notation (JSON).



### Reading JSON Files

JSON is a lightweight data interchange format. Unlike pickle, JSON is language-independent but more restricted in terms of data representation.

JSON supports the following data types:

- Atomic data types — strings, numbers, true, false, null
- Arrays — an array corresponds to a Python list; it's enclosed in square brackets []; the items in an array don't have to be of the same data type: [1, 3.14, "a string", true, null]
- Objects — an object corresponds to a Python dictionary; it is enclosed in curly braces {}; every item consists of a key and a value, separated by a colon: {"age" : 37, "gender" : "male", "married" : true}
- Any recursive combinations of arrays, objects, and atomic data types (arrays of objects, objects with arrays as item values, and so on)

## 2. Data-Set Preparation:-

There are a vast number of different types of **data preparation techniques** that could be used on a predictive modeling project.

In some cases, the distribution of the data or the requirements of a machine learning model may suggest the data preparation needed, although this is rarely the case given the complexity and high dimensionality of the data, the ever-increasing parade of new machine learning algorithms and limited, although human, limitations of the practitioner.

Instead, data preparation can be treated as another hyperparameter to tune as part of the modeling pipeline. This raises the question of how to know what data preparation methods to consider in the search, which can feel overwhelming to experts and beginners alike.

The solution is to think about the vast field of data preparation in a structured way and systematically evaluate data preparation techniques based on their effect on the raw data.

**Some RAW data examples include: -**

```
{
"intents":
    [
 {"tag": "greeting",
 "patterns": ["hi", "Is anyone there?", "hello", "Whats up", "hiiii","hai"],
"responses": ["hello!","good to see you again!", "hi there, how can I help?"]
},

{"tag": "care",
"patterns": ["how are you", "how's you"],
"responses": ["i am absoulty fine", "i am fantastic, what about you?"]
 },
.
.
.
{"tag": "phonecall",
"patterns": ["make a trail phone call", "make a phone call"],
"responses": ["phonecall"]
 }
]
}
```

## Framework for Data Preparation

Effective data preparation requires that the data preparation techniques available are organized and considered in a structured and systematic way. This allows you to ensure that approach techniques are explored for your dataset and that potentially effective techniques are not skipped or ignored.

This can be achieved using a framework to organize data preparation techniques that consider their effect on the raw dataset.

For example, structured machine learning data, such as data we might store in a CSV file for classification and regression, consists of rows, columns, and values. We might consider data preparation techniques that operate at each of these levels.

1. **Data Preparation for Rows**

2. **Data Preparation for Columns**

3. **Data Preparation for Values**

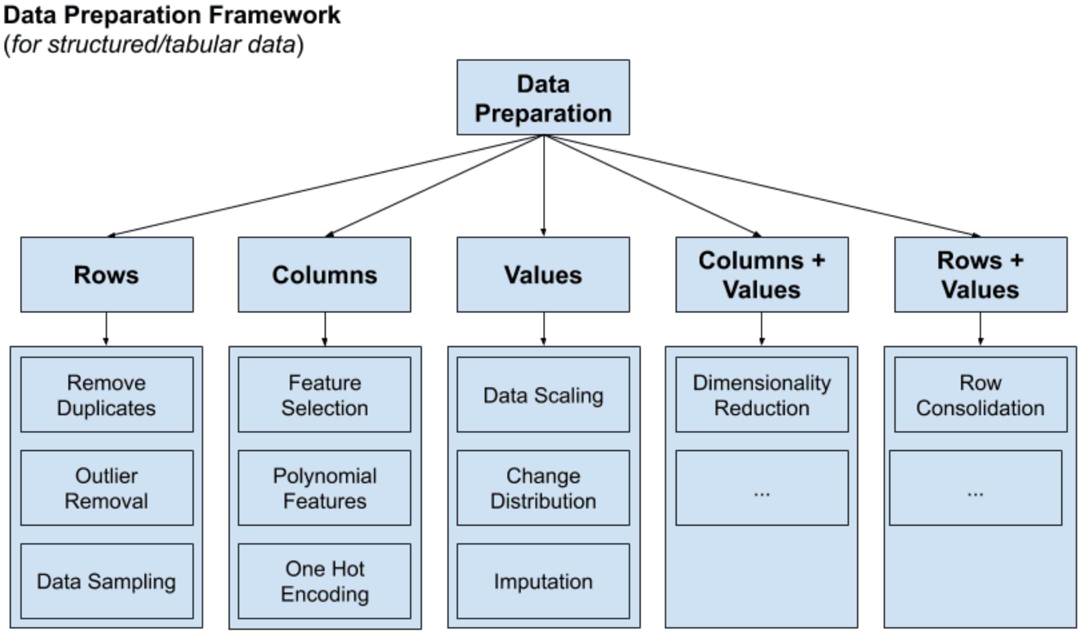
Data preparation for rows may be techniques that add or remove rows of data from the dataset. Similarly, data preparation for columns may be techniques that add or remove columns (features or variables) from the dataset. Whereas data preparation for values may be techniques that change the values in the dataset, often for a given column.

There is one more type of data preparation that does not neatly fit into this structure, and that is dimensionality reduction techniques. These techniques change the columns and the values at the same time, e.g., projecting the data into a lower-dimensional space.

Data Preparation for Columns + Values

This raises the question of techniques that might apply to rows and values at the same time. This might include data preparation that consolidates rows of data in some way.

Data Preparation for Rows + Values



## 3. Feature Extraction:-

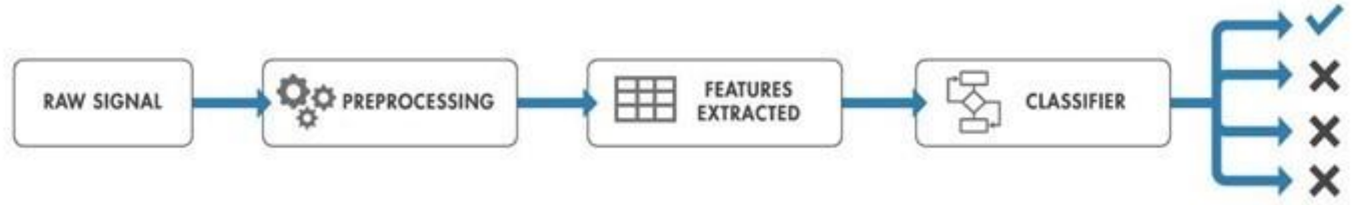
Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. It yields better results than applying machine learning directly to the raw data.

Automated feature extraction uses specialized algorithms or deep networks to extract features automatically from signals or images without the need for human intervention. This technique can be very useful when you want to move quickly from raw data to developing machine learning algorithms. Wavelet scattering is an example of automated feature extraction.

With the ascent of deep learning, feature extraction has been largely replaced by the first layers of deep networks – but mostly for image data. For signal and time-series applications, feature extraction remains the first challenge that requires significant expertise before one can build effective predictive models.

## 7.1. Feature Extraction Process

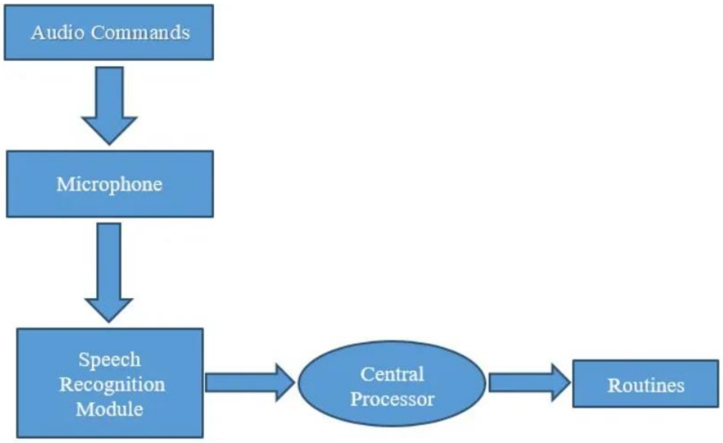
Feature extraction identifies the most discriminating characteristics in signals, which a machine learning or a deep learning algorithm can more easily consume. Training machine learning or deep learning directly with raw signals often yields poor results because of the high data rate and information redundancy.



Schematic process for applying feature extraction for a machine learning classifier.

## 7.2. Proposed Plan of Work

The work started with analyzing the audio commands given by the user through the microphone. This can be anything like getting any information, operating a computer’s internal files, etc. This is an empirical qualitative study, based on reading above mentioned literature and testing their examples. Tests are made by programming according to books and online resources, with the explicit goal of finding best practices and a more advanced understanding of Voice Assistant.



### 7.2.1. Virtual Assistant Using Python

The figure shows the workflow of the basic process of the voice assistant. Speech recognition is used to convert the speech input to text. This text is then fed to the central processor which determines the nature of the command and calls the relevant script for execution.

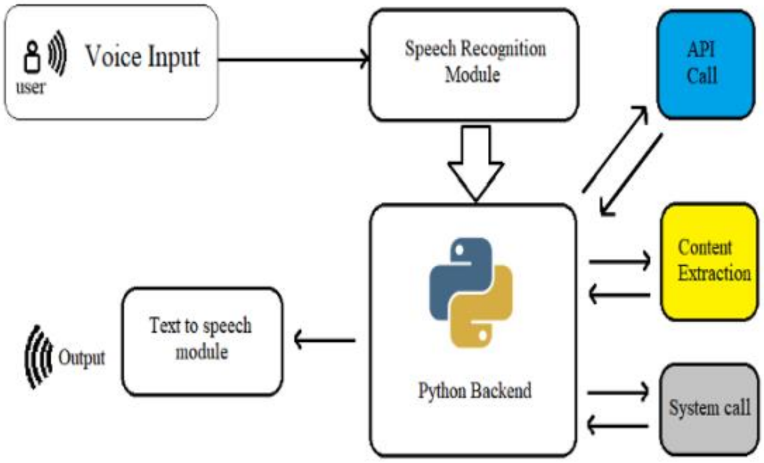
But the complexities don't stop there. Even with hundreds of hours of input, other factors can play a huge role in whether or not the software can understand you. Background noise can easily throw a speech recognition device off track. This is because it does not inherently have the ability to distinguish the ambient sounds it "hears" of a dog barking or a helicopter flying overhead, from your voice.

Engineers have to program that ability into the device; they conduct data collection of these ambient sounds and "tell" the device to filter them out. Another factor is the way humans naturally shift the pitch of their voice to accommodate for noisy environments; speech recognition systems can be sensitive to these pitch changes.

## 7.3. Methodology of Virtual Assistant Using Python

**Speech Recognition module:**

The system uses Google's online speech recognition system for converting speech input to text. The speech input Users can obtain texts from the special corpora organized on the computer network server at the information centre from the microphone is temporarily stored in the system which is then sent to Google cloud for speech recognition. The equivalent text is then received and fed to the central processor.



**Python Backend:**

The python backend gets the output from the speech recognition module and then identifies whether the command or the speech output is an API Call and Context Extraction. The output is then sent back to the python backend to give the required output to the user.

**API calls:**

API stands for Application Programming Interface. An API is a software intermediary that allows two applications to talk to each other. In other words, an API is a messenger that delivers your request to the provider that you're requesting it from and then delivers the response back to you.

**Content Extraction:**

Context extraction (CE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. In most cases, this activity concerns processing human language texts using natural language processing (NLP). Recent activities in multimedia document processing like automatic annotation and content extraction out of images/audio/video could be seen as context extraction TEST RESULTS.

**Text-to-speech module:**

Text-to-Speech (TTS) refers to the ability of computers to read text aloud. A TTS Engine converts written text to a phonemic representation, then converts the phonemic representation to waveforms that can be output as sound. TTS engines with different languages, dialects and specialized vocabularies are available through third-party publishers.

## 7.4. DETAILED ANALYSIS

In order to understand how a machine learning algorithm learns from data to predict an outcome, it is essential to understand the underlying concepts involved in training an algorithm.

## 7.4.1. Estimators

Estimation is a statistical term for finding some estimate of unknown parameter, given some data. Point Estimation is the attempt to provide the single best prediction of some quantity of interest.

Quantity of interest can be:
A single parameter
A vector of parameters — e.g., weights in linear regression
A whole function

### Point estimator

To distinguish estimates of parameters from their true value, a point estimate of a parameter θ is represented by θˆ. Let {x(1) , x(2) ,….., x(m)} be m independent and identically distributed data points. Then a point estimator is any function of the data:
This definition of a point estimator is very general and allows the designer of an estimator great flexibility. While almost any function thus qualifies as an estimator, a good estimator is a function whose output is close to the true underlying θ that generated the training data.

Point estimation can also refer to estimation of relationship between input and target variables referred to as function estimation.

### Function Estimation

Here we are trying to predict a variable y given an input vector x. We assume that there is a function f(x) that describes the approximate relationship between y and x. For example,

we may assume that y = f(x) + ε, where ε stands for the part of y that is not predictable from x. In function estimation, we are interested in approximating f with a model or estimate fˆ. Function estimation is really just the same as estimating a parameter θ; the function estimator fˆ is simply a point estimator in function space. Ex: in polynomial regression we are either estimating a parameter w or estimating a function mapping from x to y.

## 7.4.2. Bias and Variance

Bias and variance measure two different sources of error in an estimator. Bias measures the expected deviation from the true value of the function or parameter. Variance on the other hand, provides a measure of the deviation from the expected estimator value that any particular sampling of the data is likely to cause.

### Bias

The bias of an estimator is defined as:

$$\mathrm{bias}(\hat{\boldsymbol{\theta}}_m) = \mathbb{E}(\hat{\boldsymbol{\theta}}_m) - \boldsymbol{\theta}$$

where the expectation is over the data (seen as samples from a random variable)and θis the true underlying value of θused to define the data generating distribution.

An estimator θˆm is said to be unbiased if bias(θˆm) = 0, which implies that E(θˆm) = θ.

**Variance and Standard Error**

The variance of an estimator Var(θˆ) where the random variable is the training set. Alternately, the square root of the variance is called the standard error, denoted standard error SE(ˆθ). The variance or the standard error of an estimator provides a measure of how we would expect the estimate we compute from data to vary as we independently re-sample the dataset from the underlying data generating process.

**Maximum Likelihood Estimator (MLE)**

Maximum Likelihood Estimation can be defined as a method for estimating parameters (such as the mean or variance) from sample data such that the probability (likelihood) of obtaining the observed data is maximized.

Consider a set of m examples X = {x(1), . . . , x(m)} drawn independently from the true but unknown data generating distribution Pdata(x). Let Pmodel(x; θ) be a parametric family of probability distributions over the same space indexed by θ. In other words, Pmodel(x; θ) maps any configuration x to a real number estimating the true probability Pdata(x).

The maximum likelihood estimator for θ is then defined as:

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} p_{\text{model}}(\mathbb{X}; \boldsymbol{\theta})$$

Since we assumed the examples to be i.i.d, the above equation can be written in the product form as:

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \prod_{i=1}^{m} p_{\text{model}}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta})$$

This product over many probabilities can be inconvenient for a variety of reasons. For example, it is prone to numerical underflow. Also, to find the maxima/minima of this function, we can take the derivative of this function w.r.t θ and equate it to 0. Since we have terms in product here, we need to apply the chain rule which is quite cumbersome with products. To obtain a more convenient but equivalent optimization problem, we observe that taking the logarithm of the likelihood does not change its arg max but does conveniently transform a product into a sum and since log is a strictly increasing function ( natural log function is a monotone transformation), it would not impact the resulting value of θ.

So we have:

$$\boldsymbol{\theta}_{\mathrm{ML}} = \arg\max_{\boldsymbol{\theta}} \sum_{i=1}^{m} \log p_{\mathrm{model}}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}).$$

**Two important properties: Consistency & Efficiency**

Consistency: As the number of training examples approaches infinity, the maximum likelihood estimate of a parameter converges to the true value of the parameter.

Efficiency: A way to measure how close we are to the true parameter is by the expected mean squared error, computing the squared difference between the estimated and true parameter values, where the expectation is over m training samples from the data generating distribution. That parametric mean squared error decreases as m increases, and for m large, the Cramér-Rao lower bound shows that no consistent estimator has a lower mean squared error than the maximum likelihood estimator. When the number of examples is small enough to yield over-fitting behavior, regularization strategies such as weight decay may be used to obtain a biased version of maximum likelihood that has less variance when training data is limited.

**Maximum A Posteriori (MAP) Estimation**

Following Bayesian approach by allowing the prior to influence the choice of the point estimate. The MAP can be used to obtain a point estimate of an unobserved quantity on the basis of empirical data. The MAP estimate chooses the point of maximal posterior probability (or maximal probability density in the more common case of continuous θ):

$$\boldsymbol{\theta}_{\mathrm{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta} \mid \boldsymbol{x}) = \arg\max_{\boldsymbol{\theta}} \log p(\boldsymbol{x} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta})$$

Where on the right hand side, log p(x|θ)is the standard log likelihood term, and log p(θ), corresponding to the prior distribution.

As with full Bayesian inference, MAP Bayesian inference has the advantage of leveraging information that is brought by the prior and cannot be found in the training data. This additional information helps to reduce the variance in the MAP point estimate (in comparison to the ML estimate). However, it does so at the price of increased bias.

### 7.4.3. Loss Functions

In most learning networks, error is calculated as the difference between the actual output y and the predicted output ŷ. The function that is used to compute this error is known as Loss Function also known as Cost function.

Until now our main focus has been on parameter estimating via the MLE or MAP. The reason we discussed it before is that both MLE & MAP provide a mechanism to derive the loss function.

Let us see some commonly used loss functions.

Mean Squared Error (MSE): Mean Squared Error is one of the most common loss functions. MSE loss function is widely used in linear regression as the performance measure. To calculate MSE, you take the difference between your predictions and the ground truth, square it, and average it out across the whole dataset.

$$\mathrm{MSE} = \frac{1}{m}\sum_{i=1}^{m} ||\hat{y}^{(i)} - y^{(i)}||^2$$

where y(i) is the actual expected output and ŷ(i) is the model's prediction.

Many cost functions used in machine learning, including the MSE, can be derived from the MLE.

### 7.4.4. Deriving MSE from MLE

Linear regression algorithm learns to take an input x and produce an output value ŷ. The mapping from x to ŷis chosen to minimize mean squared error. But how did we choose MSE as a criterion for linear regression. Let us arrive at the solution from the point of view of maximum likelihood estimation. Instead of producing a single prediction ŷ, we now think of the model as producing a conditional distribution p(y|x).

We can model the linear regression problem as:

$$\hat{y} = f(x;\theta)$$

$$y \sim \mathcal{N}(y;\mu = \hat{y}, \sigma^2)$$

$$p(y|x;\theta) = \frac{1}{\sigma\sqrt{2\pi}}\exp(\frac{-(y-\hat{y})^2}{2\sigma^2})$$

we are assuming that y as having Normal distribution with ŷ as the mean of the distribution and the variance is fixed to some constant $\sigma^2$ chosen by the user. Normal distributions are a sensible choice for many applications. In the absence of prior knowledge about what form a distribution over the real numbers should take, the normal distribution is a good default choice.

Now going back to log likelihood defined earlier :

$$
\begin{aligned}
J &= \sum_{i=1}^{m} \log p(y|x;\theta) \\
&= \sum_{i=1}^{m} \log \frac{1}{\sigma\sqrt{2\pi}} \exp(\frac{-(y^{(i)} - y^{\hat{(i)}})^2}{2\sigma^2}) \\
&= \sum_{i=1}^{m} -\log(\sigma\sqrt{2\pi}) - \log\exp(\frac{(y^{(i)} - y^{\hat{(i)}})^2}{2\sigma^2}) \\
&= \sum_{i=1}^{m} -\log(\sigma) - \frac{1}{2}\log(2\pi) - \frac{(y^{(i)} - y^{\hat{(i)}})^2}{2\sigma^2} \\
&= -m\log(\sigma) - \frac{m}{2}\log(2\pi) - \sum_{i=1}^{m} \frac{(y^{(i)} - y^{\hat{(i)}})^2}{2\sigma^2} \\
&= -m\log(\sigma) - \frac{m}{2}\log(2\pi) - \sum_{i=1}^{m} \frac{\|y^{(i)} - y^{\hat{(i)}}\|^2}{2\sigma^2}
\end{aligned}
$$

where ŷ(i) is the output of the linear regression on the i-th input x^(i) and m is the number of the training examples. We see that first two terms are constant so basically maximizing the log-likelihood means minimizing the MSE as:

$$\nabla_\theta J = -\nabla_\theta \sum_{i=1}^{m} \frac{\|y^{(i)} - y^{\hat{(i)}}\|^2}{2\sigma^2}$$

We immediately see that maximizing the log-likelihood with respect to θ yields the same estimate of the parameters θ as does minimizing the mean squared error. The two criteria have different values but the same location of the optimum. This justifies the use of the MSE as a maximum likelihood estimation procedure.

**Cross-Entropy Loss (or Log Loss):**

Cross entropy measures the divergence between two probability distribution, if the cross entropy is large, which means that the difference between two distribution is large, while if the cross entropy is small, which means that two distribution is similar to each other.

Cross entropy is defined as:

$$H(P,Q) = -\sum_{x} P(x)\log Q(x)$$

where P is the distribution of the true labels, and Q is the probability distribution of the predictions from the model. It can be also shown that cross entropy loss can be as well derived from MLE, I will not bore you with more math.

Let us further simplify this for our model with:

N — number of observations
M — number of possible class labels (dog, cat, fish)
y — a binary indicator (0 or 1) of whether class label C is the correct classification for observation
p — the model’s predicted probability that observation

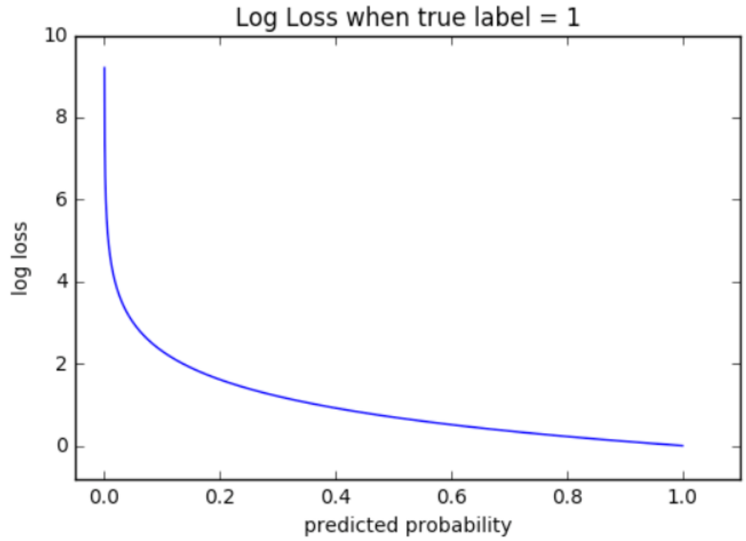
**Binary Classification:**
In binary classification (M=2), the formula equals:

$$CE = -(y \log(p) + (1 - y) \log(1 - p))$$

In case of a binary classification each predicted probability is compared to the actual class output value (0 or 1) and a score is calculated that penalizes the probability based on the distance from the expected value.

**Visualization:**

The graph below shows the range of possible log loss values given a true observation (y= 1). As the predicted probability approaches 1, log loss slowly decreases. As the predicted probability decreases, however, the log loss increases rapidly.



Log loss penalizes both types of errors, but especially those predictions that are confident and wrong!

**Multi-class Classification:**

In multi-class classification (M>2), we take the sum of log loss values for each class prediction in the observation. Cross-entropy for a binary or two class prediction problem is actually calculated as the average cross entropy across all examples. Log Loss uses negative log to provide an easy metric for comparison.

It takes this approach because the positive log of numbers < 1 returns negative values, which is confusing to work with when comparing the performance of two models. See this post for detailed discussion on cross-entropy loss.

## 7.5. ML problems and corresponding Loss functions

**Regression Problem**
A problem where you predict a real-value quantity.

- **Output Layer Configuration:** One node with a linear activation unit.
- **Loss Function:** Mean Squared Error (MSE).

**Binary Classification Problem**
A problem where you classify an example as belonging to one of two classes. The problem is framed as predicting the likelihood of an example belonging to class one, e.g. the class that you assign the integer value 1, whereas the other class is assigned the value 0.

- **Output Layer Configuration:** One node with a sigmoid activation unit.
- **Loss Function:** Cross-Entropy, also referred to as Logarithmic loss.

**Multi-Class Classification Problem**
A problem where you classify an example as belonging to one of more than two classes. The problem is framed as predicting the likelihood of an example belonging to each class.

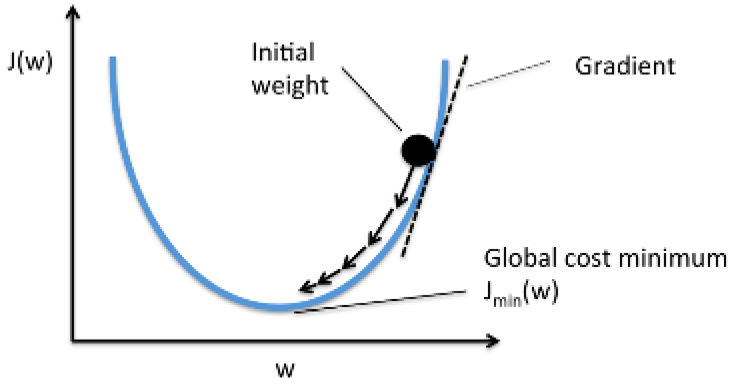
- **Output Layer Configuration:** One node for each class using the SoftMax activation function.
- **Loss Function**: Cross-Entropy, also referred to as Logarithmic loss.

**Optimizers**
To minimize the prediction error or loss, the model while experiencing the examples of the training set, updates the model parameters W. These error calculations when plotted against the W is also called cost function plot J(w), since it determines the cost/penalty of the model. So minimizing the error is also called minimization of the cost function.

The most important technique and the foundation of how we train and optimize our model is using Gradient Descent.

### 7.5.1. Gradient Descent :

When we plot the cost function J(w) vs w. It is represented as below:



As we see from the curve, there exists a value of parameters W which has the minimum cost Jmin. Now we need to find a way to reach this minimum cost.

In the gradient descent algorithm, we start with random model parameters and calculate the error for each learning iteration, keep updating the model parameters to move closer to the values that results in minimum cost.

repeat until minimum cost: {

$$w_j = w_j - \alpha\partial/\partial w_j J(W)$$

}

In the above equation we are updating the model parameters after each iteration. The second term of the equation calculates the slope or gradient of the curve at each iteration.

The gradient of the cost function is calculated as partial derivative of cost function J with respect to each model parameter Wj, j takes value of number of features [1 to n]. α, alpha, is the learning rate, or how quickly we want to move towards the minimum. If α is too large, we can overshoot. If α is too small, means small steps of learning hence the overall time taken by the model to observe all examples will be more.

There are three ways of doing gradient descent:

**Batch gradient descent:** Uses all of the training instances to update the model parameters in each iteration.

**Mini-batch Gradient Descent:** Instead of using all examples, Mini-batch Gradient Descent divides the training set into smaller size called batch denoted by ‘b’. Thus a mini-batch ‘b’ is used to update the model parameters in each iteration.

**Stochastic Gradient Descent (SGD):** updates the parameters using only a single training instance in each iteration. The training instance is usually selected randomly. Stochastic gradient descent is often preferred to optimize cost functions when there are hundreds of thousands of training instances or more, as it will converge more quickly than batch gradient descent .

Some other commonly used optimizers:

**Adagrad**
Adagrad adapts the learning rate specifically to individual features: that means that some of the weights in your dataset will have different learning rates than others. This works really well for sparse datasets where a lot of input examples are missing. Adagrad has a major issue though: the adaptive learning rate tends to get really small over time. Some other optimizers below seek to eliminate this problem.

**RMSprop**
RMSprop is a special version of Adagrad developed by Professor Geoffrey Hinton in his neural nets class. Instead of letting all of the gradients accumulate for momentum, it only accumulates gradients in a fixed window. RMSprop is similar to Adaprop, which is another optimizer that seeks to solve some of the issues that Adagrad leaves open.

**Adam**
Adam stands for adaptive moment estimation, and is another way of using past gradients to calculate current gradients. Adam also utilizes the concept of momentum by adding fractions of previous gradients to the current one. This optimizer has become pretty widespread, and is practically accepted for use in training neural nets.

## 7.6. Training with pyTorch

### 7.6.1. Dataset and DataLoader

The Dataset and DataLoader classes encapsulate the process of pulling your data from storage and exposing it to your training loop in batches.

The Dataset is responsible for accessing and processing single instances of data.

The DataLoader pulls instances of data from the Dataset (either automatically or with a sampler that you define), collects them in batches, and returns them for consumption by your training loop.

The DataLoader works with all kinds of datasets, regardless of the type of data they contain.

In this project, I choose Stochastic Gradient Descent(SGD) and Adaptive Moment Estimation (Adam) optimizer to carried out the training process of the machine using pyTorch tensor.

### 7.6.2. The Training Loop

It enumerates data from the DataLoader, and on each pass of the loop does the following:

- Gets a batch of training data from the DataLoader
- Zeros the optimizer’s gradients
- Performs an inference - that is, gets predictions from the model for an input batch
- Calculates the loss for that set of predictions vs. the labels on the dataset
- Calculates the backward gradients over the learning weights
- Tells the optimizer to perform one learning step - that is, adjust the model’s learning weights based on the observed gradients for this batch, according to the optimization algorithm we chose
- It reports on the loss for every 1000 batches.
- Finally, it reports the average per-batch loss for the last 1000 batches, for comparison with a validation run

Save a copy of the model using .pth file extension.

# CHAPTER-8

**(RESULTS)**

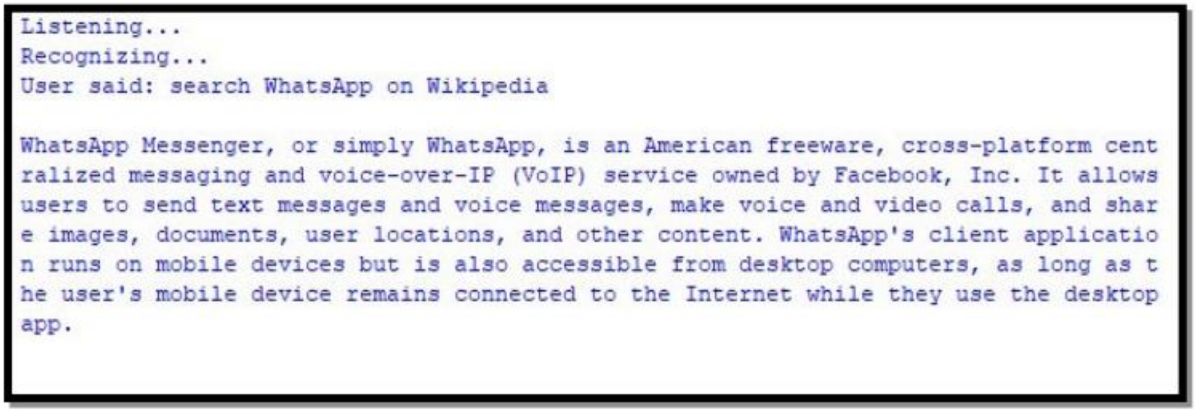
# OUTPUT

When you run the program, first the voice assistant “J.A.R.V.I.S.” greets you by saying “Good Morning/Afternoon/Evening” according to current time. After that, it will introduce itself. It will ask the user “What can I do for you?”. Now there are various commands which are completely functioning right now which are as following:-

- Search about something on Wikipedia.
- Open websites like Google.com, Youtube.com, etc.
- Open system applications.
- Weather Report.
- News Report.
- Maps.
- Writing some notes.
- And some conversation related questions.

## 8.1. Search on Wikipedia :-

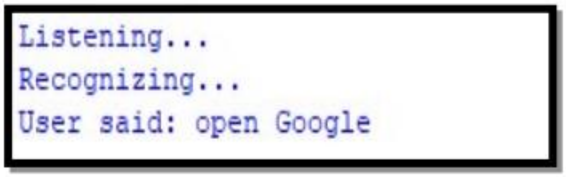
The Wikipedia search function can be used by giving command like “Search *something* on Wikipedia”. The Wikipedia module is used to execute this command. This function can be very useful to know or get information about anything from Wikipedia. Once the user gives voice command to assistant, it recognizes the keywords used in the query and process it and gives the results.



```
Listening...
Recognizing...
User said: search WhatsApp on Wikipedia

WhatsApp Messenger, or simply WhatsApp, is an American freeware, cross-platform cent
ralized messaging and voice-over-IP (VoIP) service owned by Facebook, Inc. It allows
users to send text messages and voice messages, make voice and video calls, and shar
e images, documents, user locations, and other content. WhatsApp's client applicatio
n runs on mobile devices but is also accessible from desktop computers, as long as t
he user's mobile device remains connected to the Internet while they use the desktop
app.
```

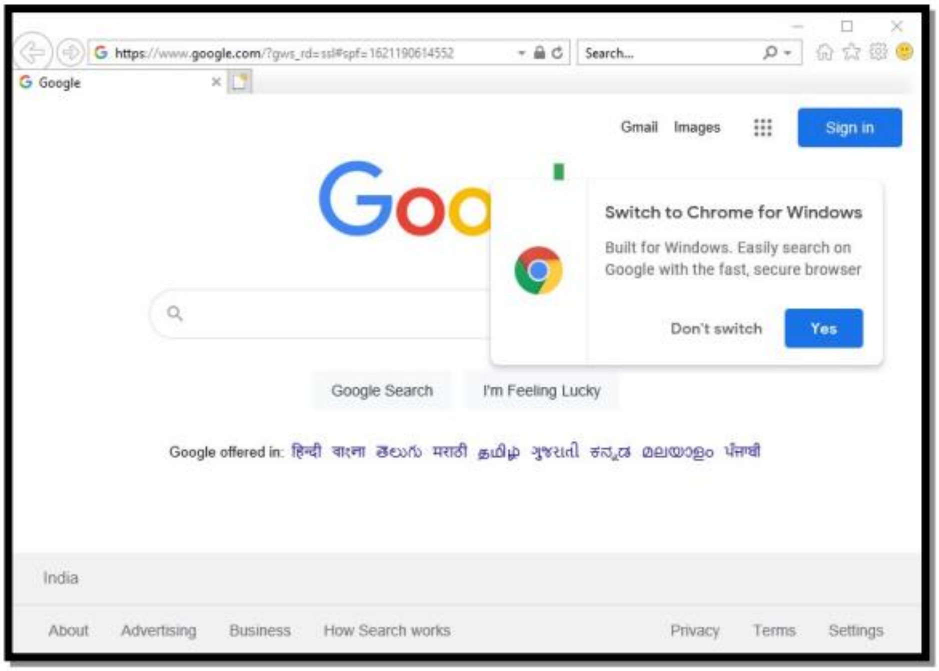
## 8.2. Open Websites on Web browsers:

To open some websites like Google.com, YouTube.com, etc. the user gives the voice command to assistant that which website he/she wants to open. For example, the user says “Open Google”, the assistant receives the voice command and process it according to the query.



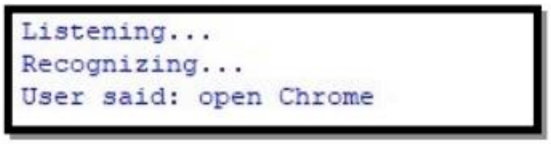
```
Listening...
Recognizing...
User said: open Google
```

The web browser module is used for this function which opens the URLs of the given website on the web browser of the system.


https://www.google.com/?gws_rd=ssl#spf=1621190614552
Search...
Google
Gmail
Images
Sign in
Switch to Chrome for Windows
Built for Windows. Easily search on Google with the fast, secure browser
Don't switch
Yes
Google Search
I'm Feeling Lucky
Google offered in: हिन्दी বাংলা తెలుగు मराठी தமிழ் ગુજરાતી ಕನ್ನಡ മലയാളം ਪੰਜਾਬੀ
India
About
Advertising
Business
How Search works
Privacy
Terms
Settings

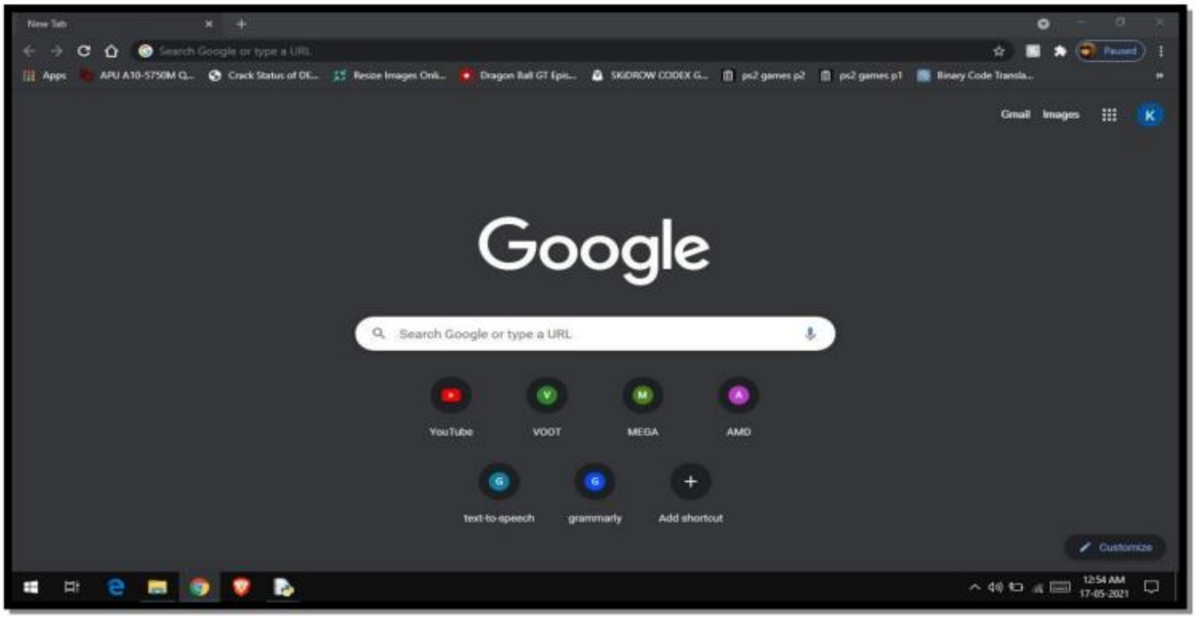
## 8.3. Run system applications:-

Opening some system applications is very easy by using voice command. The user just gives voice command like “open *application name*” and the voice assistant will open that application for you.



```
Listening...
Recognizing...
User said: open Chrome
```

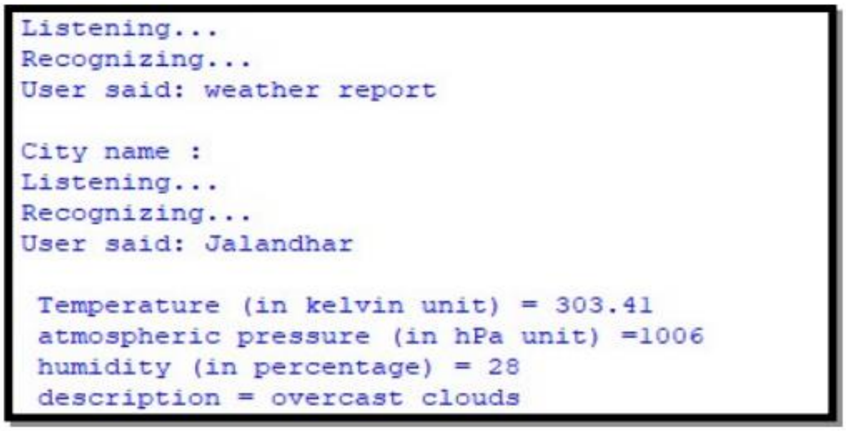
For example, the user gives voice command for open Chrome application. The voice assistant will go to the file location of that application and simply run that .exe file of that application.



## 8.4. Weather Report:-

The weather report function can be used by using OpenWeatherAPI module. OpenWeatherMap is an online service which is used to know the weather forecast of the particular city. To use this service, the user needs to create an account first and they gives the API ID for that account.

This is an online service used to learn about the city's weather forecast. The user must first create an account in order to use this service and then provide the API ID for that account.

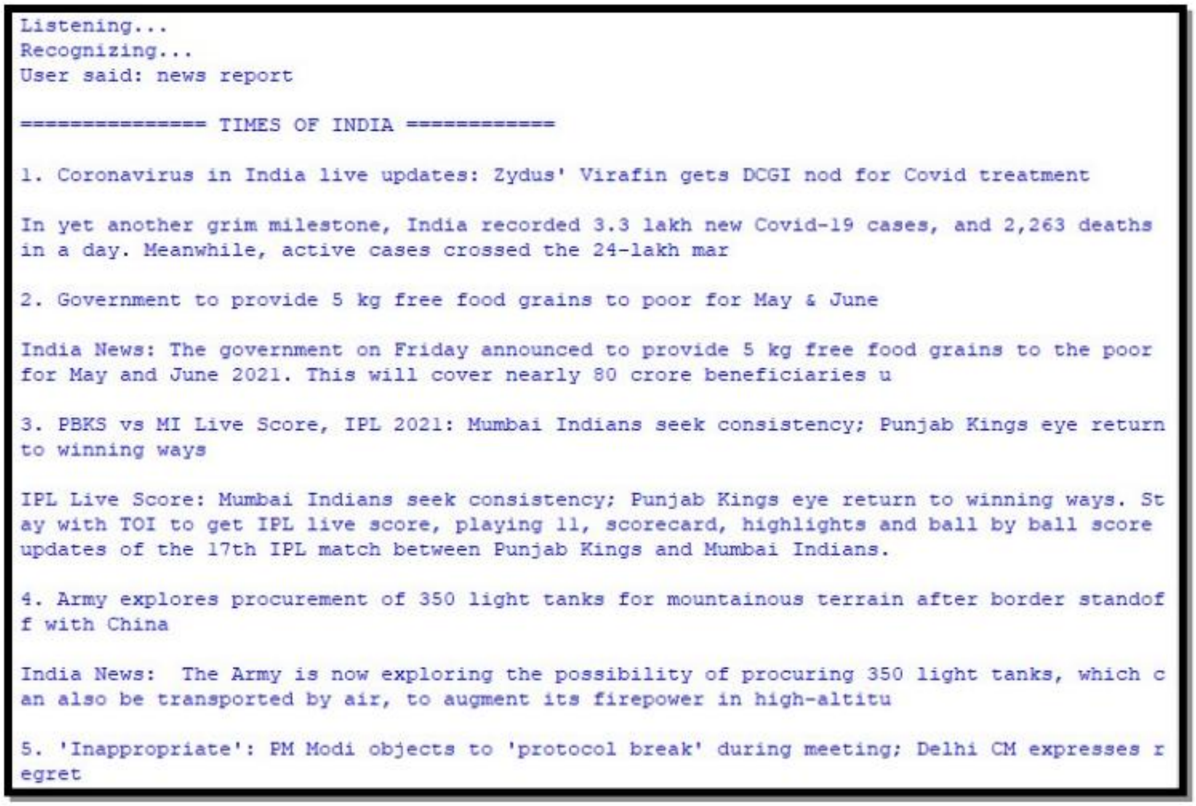


```
Listening...
Recognizing...
User said: weather report

City name :
Listening...
Recognizing...
User said: Jalandhar

 Temperature (in kelvin unit) = 303.41
 atmospheric pressure (in hPa unit) =1006
 humidity (in percentage) = 28
 description = overcast clouds
```

The user gives the voice command “Weather report” to the assistant which then ask the city name which the user wants to get weather report. Once the user gives the city name, the assistant uses the OpenWeatherMap service to get the weather forecast information of that city and displays the weather report.

## 8.5. News Report:-

The news report function can be used by using json module and NewsAPI service. To use the NewsAPI, the user needs to create an account on newsapi.org to get this service.

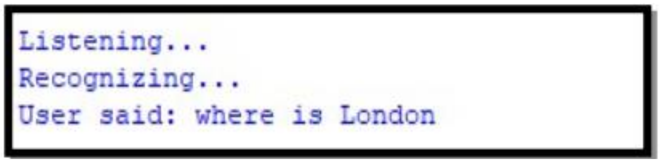


```
Listening...
Recognizing...
User said: news report

=============== TIMES OF INDIA ============

1. Coronavirus in India live updates: Zydus' Virafin gets DCGI nod for Covid treatment

In yet another grim milestone, India recorded 3.3 lakh new Covid-19 cases, and 2,263 deaths
in a day. Meanwhile, active cases crossed the 24-lakh mar

2. Government to provide 5 kg free food grains to poor for May & June

India News: The government on Friday announced to provide 5 kg free food grains to the poor
for May and June 2021. This will cover nearly 80 crore beneficiaries u

3. PBKS vs MI Live Score, IPL 2021: Mumbai Indians seek consistency; Punjab Kings eye return
to winning ways

IPL Live Score: Mumbai Indians seek consistency; Punjab Kings eye return to winning ways. St
ay with TOI to get IPL live score, playing 11, scorecard, highlights and ball by ball score
updates of the 17th IPL match between Punjab Kings and Mumbai Indians.

4. Army explores procurement of 350 light tanks for mountainous terrain after border standof
f with China

India News:  The Army is now exploring the possibility of procuring 350 light tanks, which c
an also be transported by air, to augment its firepower in high-altitu

5. 'Inappropriate': PM Modi objects to 'protocol break' during meeting; Delhi CM expresses r
egret
```

The user gives the voice command “News report” to the assistant which then reads the news headlines of the particular newspaper with the help of json module using NewsAPI service. In this project, we used Times of India newspaper.

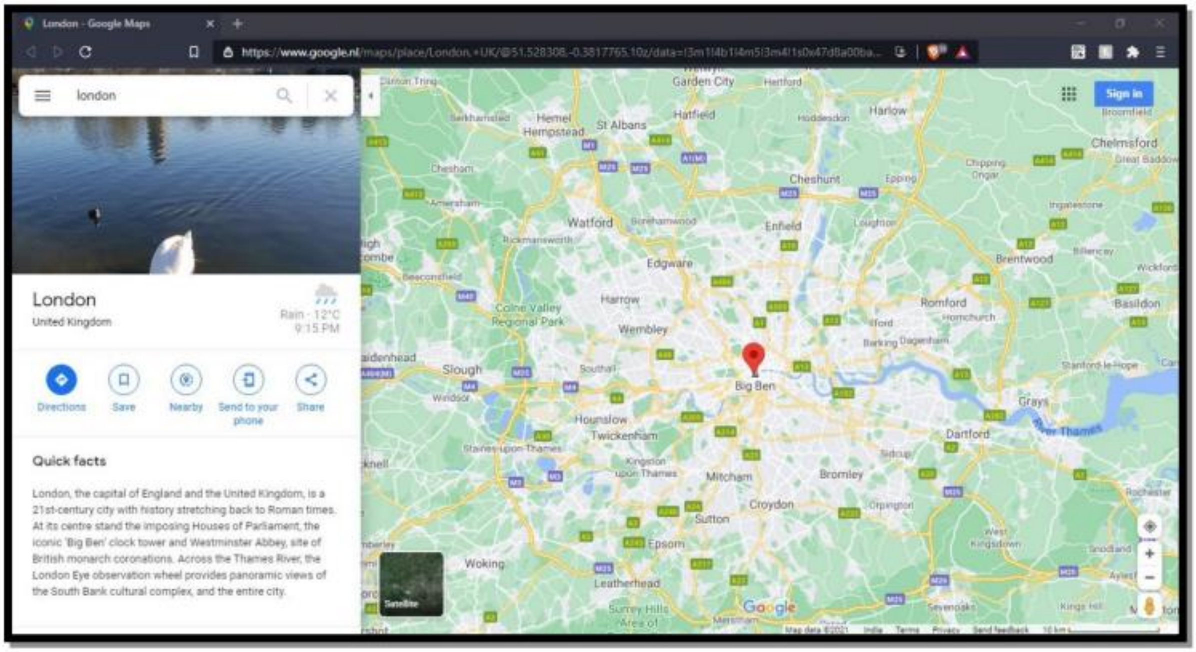
## 8.6. **Maps**:-

The location search command is very useful to find where the location is on map. This function can be executed by using Google Maps.



```
Listening...
Recognizing...
User said: where is London
```

The user gives the voice command “where is *city name or something*” to the assistant by which assistant then executes the query by locating that location with the help of Google Maps.



Quick facts

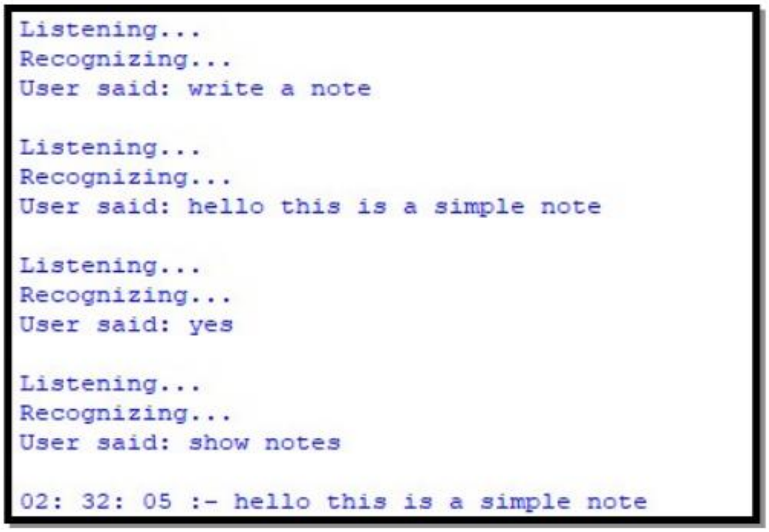
London, the capital of England and the United Kingdom, is a 21st-century city with history stretching back to Roman times. At its centre stand the imposing Houses of Parliament, the iconic 'Big Ben' clock tower and Westminster Abbey, site of British monarch coronations. Across the Thames River, the London Eye observation wheel provides panoramic views of the South Bank cultural complex, and the entire city.

## 8.7. Writing a Note:-

Writing a note using voice command is very easy than typing. To write a note, the user gives voice command “write a note” to voice assistant. After that it starts listening the voice of the user to convert the speech into text by using Text to Speech feature.

It's very easy to write a note using the voice command. The user gives a 'write a note' Voice Assistant command to write a note. Afterwards it begins to hear the user's voice to convert the speech to text using Text-to-Speech.

After writing the note, the assistant adds the date and time to that note. The note is saved as in the .txt file format.



```
Listening...
Recognizing...
User said: write a note

Listening...
Recognizing...
User said: hello this is a simple note

Listening...
Recognizing...
User said: yes

Listening...
Recognizing...
User said: show notes

02: 32: 05 :- hello this is a simple note
```

# CHAPTER-9

## (CONCLUSION & FUTURE ENHANCEMENT)

# CONCLUSION

JARVIS is a very helpful voice assistant without any doubt as it saves time for the user by conversational interactions, its effectiveness and efficiency. But while working on this project, there were some limitations encountered and also realized some scope of enhancement in the future which are mentioned below:

## 9.1. LIMITATIONS:-

- Security is somewhere an issue, there is no voice command encryption in this project.
- Background voice can interfere
- Misinterpretation because of accents may cause inaccurate results.
- JARVIS cannot be called externally anytime like other traditional assistants like Google Assistant can be called just by saying, "Ok Google!"

## 9.2. SCOPE FOR FUTURE WORK:-

- Make JARVIS to learn more on its own and develop a new skill in it.
- JARVIS android app can also be developed.
- Make more Jarvis voice terminals.
- Voice commands can be encrypted to maintain security.
- Strip back the physical hardware as far as possible.
- With the intelligence of the Virtual Personal Assistant, getting pulled in, and pushing its way into our lives on multiple devices on our bodies and in our homes, workplaces and vehicles.
- Your Virtual Personal Assistant will be continually prompting you with suggestions and taking instructions and will know more about you than perhaps you do yourself.
- We can expect this device to be implanted and permanent.

# (BIBLIOGRAPHY)

# REFERENCES

- Cortana Intelligence, Google Assistant, Apple Siri
- In Paper [2.1], the author being prepared on Virtual Agent Behaviour. The author are Maaike Harbers1,2, Karel van den Bosch, and John-Jules Meyer, 1 Utrecht University, P.O.Box 80.089, 3508 TB Utrecht, The Netherlands{maaike,jj}@cs.uu.nl 2 TNO Human Factors, P.O.Box 23, 3769 ZG Soesterberg, The Netherlandskarel.vandenbosch@tno.nl.
- In Paper [2.2], the author being prepared on Speech Recognition. The author are Nitin Washani, M. Tech Scholar, DIT University, Dehradun (Uttarakhand, India), Sandeep Sharma, Ph.D., Head of Department of ECE DIT University, Dehradun (Uttarakhand, India).
- In Paper [2.3], the author being prepared on Natural Language Processing. The author are Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh, Department of Computer Science and Engineering Manav Rachna International University, Faridabad-121004, India, Accendere Knowledge Management Services Pvt. Ltd., India.
- In Paper [2.4], the author being prepared on Neural Network. The author are Maad M. Mijwel, Adam Esen, Aysar Shamil, Computer Engineering Techniques Department, Baghdad College of Economics Sciences University, Iraq, Social Sciences Institute Economic Law, University of Gaziantep, Turkey, Information technology faculty, Middle East University, Jordan.
- In Paper [2.5], the author being prepared on Cross Entropy Error. The author are G.E. Nasr, E.A. Badr and C. Joun, School of Engineering and Architecture, Lebanese American University, Byblos, Lebanon, e-mail: genasr@lau.edu.lb.
- In Paper [2.6], the author being prepared on Optimizers in Neural Network. The author is Chitra Desai Professor Department of Computer Science National Defence Academy, Pune, India.
- In Paper [2.7], the author being prepared on Strength Prediction with ANN. The author are Meltem Ozturan ,Birgul Kutlu , Turan Ozturan.
- In Paper [2.8], the author being prepared on Qt Designer IDE. The author are D. Berton.
- In Paper [2.9], the author being prepared on NumPy. The author are Ekaba Bisong, SiliconBlast.
- In Paper [2.10], the author being prepared on PyTorch. The author are Sagar Imambi, Bhanu Prakash Kolla, K L University, G.R. Kanagachidambaresan, Vel Technical University.

- In Paper [2.11], the author being prepared on APIs. The author are Sergio Inzunza, Autonomous University of Baja California, Reyes Juarez-Ramirez, Autonomous University of Baja California, Samantha Jimenez, Autonomous University of Baja California.

- In Paper [2.12], the author being prepared on Modules and Packages in Python. The author are John Hunt.

- https://data-flair.training/blogs/artificial-intelligence-project-ideas/

- https://www.upgrad.com/blog/top-artificial-intelligence-project-ideas-topics-for-beginners/

- https://www.activestate.com/blog/how-to-build-a-digital-virtual-assistant-in-python/

- https://medium.com/codex/making-your-own-ai-virtual-assistant-with-python-5c2046dadfa7

- https://nevonprojects.com/voice-based-intelligent-virtual-assistance-for-windows/

- https://www.codewithharry.com/

- Python Programming – Kiran Gurbani

- Learning Python - Mark Lutz

- Designing Personal Assistant Software for Task Management using Semantic Web Technologies and Knowledge

- Python code for Artificial Intelligence: Foundations of Computational Agents David L. Poole and Alan

- How to use an API with Python(Beginner's Guide). Retrieved from the link

- https://rapidapi.com/blog/how-to-use-an-api-with-python/

- Introduction to Machine Learning using Python (January 2019). Retrieved from

- https://www.geeksforgeeks.org/introduction-machine-learning-using-python/

- Swapnil Saurav, Python Programing-learn and practice (2nd Edition). Ingram short title; 2018 January

- "Study of Voice Controlled Personal Assistant Device" by IJCTT Volume 42 Number 1 – December 2016