CHAPTER ONE

INTRODUCTION

1.1 BACKGROUND TO THE STUDY

AI is a versatile field, and it must be effectively directed, supervised, and controlled to ensure optimal results.

On the aspect of results, biases are in AI and ML models. That is why experts must consider fairness, quality, and transparency when developing machine learning models to combat bigotry and mitigate drift.

The accessibility improvements alone are worth considering. Speech recognition allows the elderly and the physically and visually impaired to interact with state-of-the-art products and services quickly and naturally—no GUI needed!

Speech recognition has its roots in research done at Bell Labs in the early 1950s. Early systems were limited to a single speaker and had limited vocabularies of about a dozen words. Modern speech recognition systems have come a long way since their ancient counterparts. They can recognize speech from multiple speakers and have enormous vocabularies in numerous languages.

The first component of speech recognition is, of course, speech. Speech must be converted from physical sound to an electrical signal with a microphone and digital data with an analog-to-digital converter. Once digitized, several models can be used to transcribe the audio to text

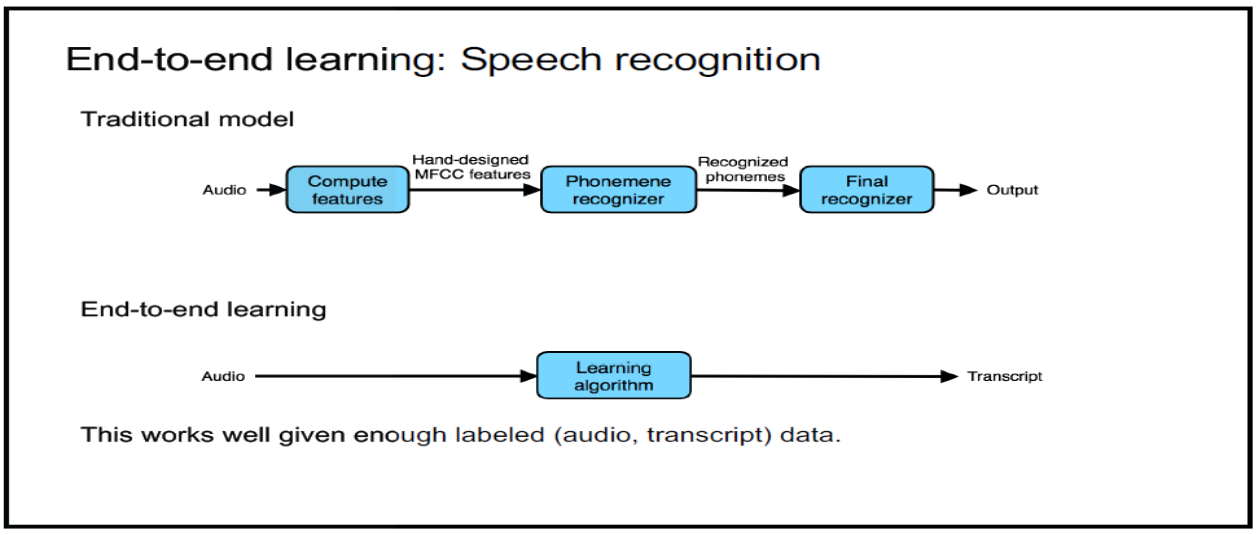


Figure1 **SPEECH RECOGNITION PHASE**

Most modern speech recognition systems rely on what is known as a **Hidden Markov Model** (HMM)*. This approach works on the assumption that a speech signal when viewed on a short enough timescale (say, ten milliseconds), can be reasonably approximated as a stationary process—that is, a process in which statistical properties do not change over time.

In a typical HMM, the speech signal is divided into 10-millisecond fragments. The power spectrum of each segment, which is essentially a plot of the signal's power as a function of frequency, is mapped to a vector of real numbers known as **cepstral** coefficients. The dimension of this vector is usually tiny—sometimes as low as 10, although more accurate systems may have dimension 32 or more. The final output of the HMM is a sequence of these vectors.

To decode the speech into text, groups of vectors are matched to one or more **phonemes**—a fundamental speech unit. This calculation requires training since the sound of a phoneme varies from speaker to speaker, differs from one utterance to another by the same speaker. A special algorithm is then applied to determine the most likely word (or words) that produce the given sequence of phonemes.

One can imagine that this whole process may be computationally expensive. In many modern speech recognition systems, neural networks are implemented to simplify the speech signal using feature transformation and dimensionality reduction techniques *before* HMM recognition. Voice activity detectors (VADs) are also used to reduce an audio signal to only the portions likely to contain speech. This prevents the recognizer from wasting time analyzing unnecessary parts of the signal.

Fortunately, as a Python programmer, you do not have to worry about any of this. Some speech recognition services are available for use online through an API, and many of these services offer **Python SDKs.**

Speech recognition technologies also use **Natural Language Processing(NLP),** meaning words converted from speech. NLP can also reach an accuracy of 96-99%, constantly improving its comprehension address with the help of users through ***FINE-TUNING*** and ***ACTIVE LEARNING*** as speech recognition technology improves with more data, as do ***VOICE USER INTERFACES (VUI)*** and user experience.

ML/AI should be used to solve this task as it requires lots and lots of voice data, probably **Gigabytes** or **Terabytes** of data.

More data is proportional to more bias. Hence, data collected from various countries would make the model perform well with data it infrequently sees, making a massive purchase or use of voice recognition systems/devices in companies, offices, and various homes.

Therefore, only ML/AI algorithms are suitable when dealing with this spectrum of data.

1.1.1 TYPES OF BIASES

1. **Algorithmic bias**

   Algorithmic bias is the error that occurs when the algorithm at the core of the machine learning process is faulty or inappropriate for the current application. Algorithmic bias can be spotted when the application gives wrong results (output cases) for a specific group of people (input cases).

   Algorithmic bias may be intentional or unintentional.

   It could be the result of technical issues within the core of the algorithm or a wrongful choice of algorithms in the first place.

2. **Sample bias**

   Sample bias results from an error in the early stages of application development, which is the collection and cleaning of data.

   Data is the core of any Machine Learning application; after all, the algorithm can't learn from what it has not been fed.

   If a developer chose a wrong sample, one that is small in size, or contains many faulty data points, or doesn't represent the entire data pool, to train their model, the results will be inaccurate for data points that differ from this sample

   .

   Luckily, sample bias is not that complex to fix; using a larger, more diverse dataset to train the model and fine-tuning parameters would provide the best answer.

3. **Prejudice bias**

   Prejudice bias is often the result of the data being biased in the first place.

   The data extracted and used to train a model may have preexisting biases, such as stereotypes and faulty case assumptions.

   So, using this data will always provide biased results no matter the algorithm used.

   Prejudice bias is quite difficult to solve; you can try to use an entirely new dataset and modify the data to eliminate any existing discrimination.

   For example, the algorithmic Justice League's voice erasure project recently found that speech recognition systems from Apple, Amazon, Google, IBM, and Microsoft collectively achieve word error rates of 35% for African American voices versus 19% for White agents.

4. **Measurement bias**

   In this bias, if the data that the model's performance and accuracy entirely depend on is inaccurate, nothing in the process remaining steps will be done.

   This data is often of some computations and measurements done either by a human or a computer and stored in a database. If these computational measurements are faulty, they will result in erroneous data points affecting what will be fed to the model to train and develop it.

5. **Exclusion bias**
   Exclusion bias results when vital data points are excluded from the training dataset, and hence, the resulting model doesn't consider them.

1.1.2 MITIGATION OF BIASES

The key to successfully mitigating bias is first to understand how and why it occurs. Humans play a crucial role in every aspect of AI and ML, from data assembly and annotation to algorithm development and beyond. This means that AI systems always contain a degree of human error. Ignoring this reality puts any organization at risk.

Navigating bias is a challenge for all data science teams, even the most experienced. But data scientists must remain vigilant in the fight to protect the integrity of their data.

1.2 STATEMENT OF THE PROBLEM

Figure 2

Some of the weaknesses associated with IoT devices are:

First, the existing system is not efficient to use with African individuals who are visually impaired.

Second, many Africans cannot operate speech systems that have cumbersome GUIs.

Lastly, Africa does not have an API for African languages.

This project aims to mitigate bias in ethnicity and close the gap between Speech recognition IoT(systems) and African purchase decisions.

1.3 RESEARCH QUESTIONS

The research questions will be clarified in the study:

1. What is the industry problem I am trying to solve?
2. Why use ML/AI in solving this task?
3. If I am labeling images, how will this help the business?
4. What business metrics will you apply to determine the success of the product?
5. Considering the size, source of data, what biases are built into the data, and how might the data be improved?
6. What labels did I decide to add to the data?
7. And why did I choose these labels versus any other option?
8. What will the minimum viable product look like? (With sketches)
9. What persona am I designing for?
10. How will this be adopted?
11. How will I improve the product in the long term?
12. What does the go-to-market plan look like?
13. How might real-world data be different from the training data?
14. Will the product learn from new data?
15. How will I employ A/B testing to improve the product?
16. How do I plan to monitor or mitigate unwanted bias in the model?

1.4 OBJECTIVE OF THE STUDY

The primary objective of this project is to mitigate bias in Speech Recognition Systems'

Moreover, specific objectives are to:

1. Build a model with western English voices that would be helpful to tech companies producing Speech Recognition systems as it would complement their prebuilt systems.
2. Compare the performance of the proposed model vis-a-vis other related models in the literature.
3. I am using A/B testing and Versioning to improve the product.
4. Increase African purchasing decisions and satisfaction using SR systems.

1.5 RESEARCH HYPOTHESIS

H0: Model would recognize said utterance in the said system as soon as at least a part of spoken utterance has been received

H1: Model would decide based on an estimate for said utterance as soon as a predetermined level of confidence for the stated estimate has been reached, even if the spoken statement has not yet been wholly uttered.

1.6 SCOPE OF STUDY

The scope of this system is comprehensive in terms of other speech systems. With the required software for collecting voice data and providing results, the system can be implemented in the following sectors targeting visually impaired personas, children, teens, and adults:

- Educational sector
- Tech companies(automobiles)
- Military
- Medical sector

1.7 LIMITATION OF WORK

One of the limitations set before developing this system is that it uses an online mobile application approach. This is because the requirement of this system fulfills the user accessibility and makes this proposed system more convenient to use. Besides that, this proposed system provides only brief choices and short feedbacks.

1.8 PROJECT PLANNING AND SUMMARY

The project schedule is an important aspect to be considered before the project start. It is a guideline to complete and finish the project on time has given. This system takes approximately four months to develop after passing through several major phases in the system's development phase.

This chapter briefly explains steps taken, with thought-provoking questions on building a functional speech recognition system,

Chapter two will focus on a literature review that defines and identifies various types of Speech Recognition, the programming language used, and the contribution of the proposed system.

Chapter three will describe how to manage the product from the startup stage till the product scaling.

Chapter four will highlight the results of the product management using various performance metrics hence, stating the models' efficiency and comparing the results with previous related works in the literature.

Lastly, chapter five will include a summary with a precise and cogent conclusion to address the study's objectives and recommendations to support the analysis.

CHAPTER TWO
LITERATURE REVIEW

2.1 INTRODUCTION

This chapter contrasts a previous speech recognition system with an API-based system and its issues and controversies.

The review will also give a theoretical base for the study and help determine the nature of the system. In providing the basis for understanding the concept of speech recognition systems, many models, are particularly bearing the complexity of understanding spoken words by machines, are considered, among other findings in the literature.

Thiang et al. (2011) presented speech recognition using Linear Predictive Coding (LPC) and Artificial Neural Network (ANN) for controlling the movement of mobile robots. Input signals were sampled directly from the microphone, and then the extraction was done by LPC and ANN [39]. Ms.Vimala.C and Dr.V.Radha (2012) proposed an independent, isolated speech recognition system for the Tamil language. Feature extraction, acoustic model, pronunciation dictionary, and language model were implemented using HMM, which produced 88% of accuracy in 2500 words [29]. Cini Kurian and KannanBalakrishnan (2012) found the development and evaluation of different acoustic models for Malayalam continuous speech recognition. In this paper, HMM is used to compare and evaluate the Context-Dependent (CD), Context Independent (CI) models, and Context-Dependent tied (CD tied) models from this CI model 21%. The database consists of 21 speakers, including ten males and 11 females [7]. Suma Swamy et al. (2013) introduced an efficient speech recognition system which was experimented with Mel Frequency Cepstrum Coefficients (MFCC), Vector Quantization (VQ), HMM, which recognize the speech with 98% accuracy. The database consists of five words spoken by four speakers at ten times [35]. AnnuChoudhary et al. (2013) proposed an automatic speech recognition system for isolated and connected Hindi language words by using Hidden Markov Model Toolkit (HTK). Hindi words are used for the dataset extracted by MFCC, and the recognition system achieved 95% accuracy in isolated words and 90% in related words [3]. Preeti Saini et al. (2013) proposed Hindi automatic speech recognition using HTK. Secret words are used to recognize the speech with ten states in HMM topology, producing 96.61% [31]. Md. Akkas Ali et al. (2013) presented an automatic speech recognition technique for Bangla words. Feature extraction was done by Linear Predictive Coding (LPC) and Gaussian Mixture Model (GMM). One hundred words were recorded 1000 times which gave 84% accuracy [25]. Maya Moneykumar et al. (2014) developed Malayalam word identification for a speech recognition system. The proposed work was done with syllable-based segmentation using HMM on MFCC for feature extraction [24]. Jitendra Singh Pokhariya and Dr. Sanjay Mathur (2014) introduced Sanskrit speech recognition using HTK. MFCC and two states of HMM were used for extraction, producing 95.2% to 97.2% accuracy, respectively [16]. In 2014, GeetaNijhawan et al. developed a real-time speaker recognition system for Hindi words. Feature extraction is done with MFCC. Quantization Linde, Buzo, and Gray (VQLBG) algorithm. Voice Activity Detector (VAC) was proposed to remove the silence [10]

2.2 WHAT WE KNOW ABOUT SPEECH RECOGNITION

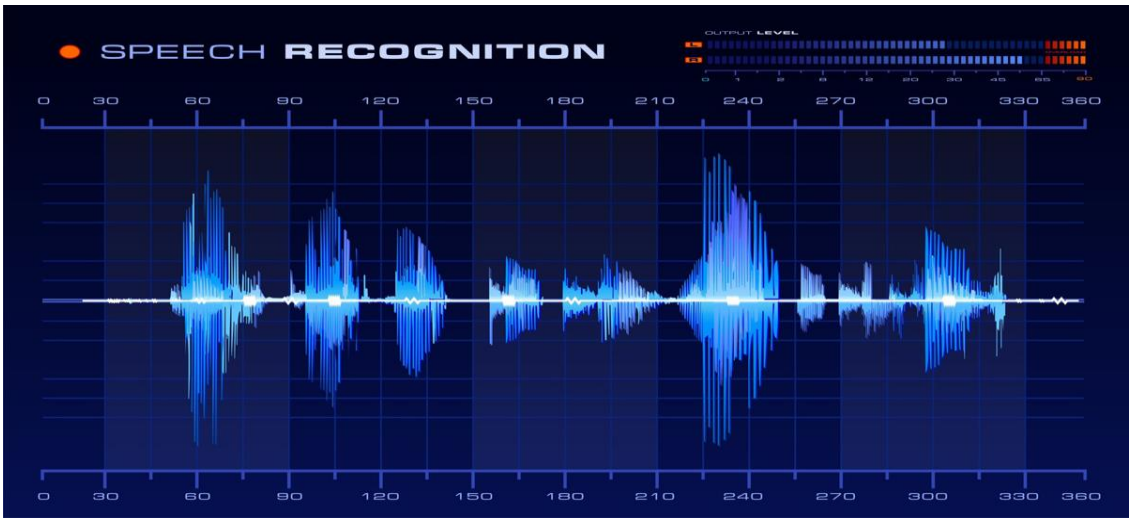


Figure 3 ***SPEECH RECOGNITION***

Speech recognition is a subfield of ***computer science*** and ***computational linguistics*** that develops methodologies and technologies that enable the recognition and translation of spoken language into text by computers.

It incorporates knowledge and research in the following areas:

- *Computer science*
- *Linguistics*
- *Computer engineering*

Speech recognition, also known as automatic speech recognition (**ASR**), computer speech recognition, or **speech-to-text**, is a capability that enables a program to process human speech into a written format. While it's commonly confused with voice recognition, speech recognition focuses on translating speech from a verbal form to a text one, whereas voice recognition seeks to identify an individual user's voice. ***(Definition from IBM)***

Recognizing the speaker can simplify the task of translating speech in systems that have been trained on a specific person's voice, or it can be used to authenticate or verify the identity of a speaker as part of a security process.

Some speech recognition systems require *"training"* (also called *"enrollment")* where an individual speaker reads text or isolated *vocabulary* into the system. The system analyzes the person's specific voice and uses it to fine-tune the recognition of that person's speech, resulting in increased accuracy. Systems that do not use training are called *"speaker-independent"* systems. Systems that use training are called *"speaker-dependent."*

From the technology perspective, speech recognition has a long history with several waves of significant innovations. Most recently, the field has benefited from advances in *deep learning* and *big data*. The passages are evidenced not only by the surge of academic papers published in the area but, more

importantly, by the worldwide industry adoption of a variety of deep learning methods in designing and deploying speech recognition systems.

2.3 HISTORY OF SPEECH RECOGNITION

In 1952, the Audrey system designed at Bell Laboratories was the first speech recognition system that recognized only digits spoken by a single person. Ten years later, IBM produced in 1962, which identified 16 English words. In collaboration, the Soviet Union, United States, England and Japan developed hardware that recognized4 vowels and nine consonants. Carnegie Mellon's "Harpy" speech-understanding system recognized 1011 words between 1971 and 1976. Threshold Technology and Bell Laboratories are the first commercial speech recognition companies that interpret multiple-person voices. A new statistical method called Hidden Markov Model (HMM) was introduced in 1980, which expanded to recognize hundreds of words to several thousand words and an unlimited number of words. Children could train to respond to their voice in the form of Worlds of Wonder's Julie dollin 1987. In 1985, Kurzweiltext-to-speech recognized 5000-word vocabulary, which IBM established.

Dragon launched the first consumer speech recognition product called Dragon Dictate, which recognizes 100 words per minute, and also, the system took 45 minutes to train the program. In 1996, Voice-Activated Link (VAL) from Bell South launched a dial-in interactive voice recognition system that gave the information based on what the speaker said through the phone. In 2001, speech recognition system attained 80% accuracy [14].Ten years later, Google's English Voice Search system integrated230 billion words from an actual user. In 2015, Google's speech recognition experimented with Connectionist Temporal Classification (CTC) trained Long Short-Term Memory(LSTM) approaches which are implemented in Google Voice [12]. Various techniques suggested by many researchers for developing different applications in speech recognition are elaborated in this paper. This article is organized as follows. Section 2 presents a classification of the speech recognition system. Section 3 analyses the related work carried out in the area of the speech recognition system. Section 4 explains the methodologies used in speech recognition. Section 5 presents the conclusion and future extension of the research work.

2.4 PYTHON FOR SPEECH RECOGNITION

Python's sizeable standard library commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as ***MIME* and *HTTP*** are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary-precision decimals, manipulating regular expressions, and unit testing.

Specifications cover some parts of the standard library, but most modules are not. They are specified by their code, internal documentation, and test suites. However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of November 2019, the ***Python Package Index (PyPI)***, the official repository for third-party Python software, contains over 200,000packages with a wide range of functionality.

A handful of packages for speech recognition exist on PyPI. A few of them include:

- apiai
- assemblyai
- google-cloud-speech
- pocketsphinx
- SpeechRecognition
- Watson-developer-cloud

Some of these packages—such as wit and apiai—offer built-in features, like ***natural language processing*** for identifying a speaker's intent, which go beyond essential speech recognition. Others, like google-cloud-speech, focus solely on speech-to-text conversion.

There is one package that stands out in terms of ease-of-use: SpeechRecognition.

Recognizing speech requires audio input, and SpeechRecognition makes retrieving this input easy. Instead of having to build scripts for accessing microphones and processing audio files from scratch, SpeechRecognition will have you up and running in just a few minutes.

The SpeechRecognition library acts as a wrapper for several famous speech API's and is thus highly flexible. One of these—the Google Web Speech API—supports a default API key hard-coded into the SpeechRecognition library. That means you can get off your feet without having to sign up for a service.

The flexibility and ease of use of the SpeechRecognition package make it an excellent choice for any Python project. However, support for every feature of each API it wraps is not guaranteed. You will need to spend some time researching the available options to find out if SpeechRecognition will work in your particular case.

2.4.1 PYTHON DESIGN PHILOSOPHY AND FEATURES

Python uses *dynamic typing* and a combination of *reference counting* and a cycle-detecting garbage collector for *memory management*. It also features dynamic *name resolution (late binding)*, which binds method and variable names during program execution.

Python's design offers some support for functional programming in the Lisp tradition. It filters, maps, and reduces functions, list comprehensions, dictionaries, sets, and generator expressions.

The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

The language's core philosophy is summarized in the document "***The Zen of Python* (*PEP 20*)**" which includes aphorisms such as:

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Readability counts.

A typical neologism in the Python community is ***pythonic,*** which can have a wide range of meanings related to program style. To say that code is pythonic is to say that it uses Python idioms well, is natural or shows fluency in the language, and conforms with Python's minimalist philosophy and emphasis on readability.

In contrast, code that is difficult to understand or reads like a rough transcription from another programming language is called ***unpythonic*.**

2.4.2 PYTHON METHODS

Methods on objects are functions attached to the object's class; Python methods have an explicit "self" parameter to access instance data, in contrast to the implicit `self` (or `this`) in some other object-oriented programming languages (e.g., C++, Java, Objective-C, or Ruby).

2.4.3 PYTHON LIBRARIES

Python's extensive *standard library*, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many common formats and protocols such as *MIME* and *HTTP* are supported. It includes modules for creating *graphical user interfaces*, connecting to *relational databases*, *generating pseudorandom numbers*, arithmetic with arbitrary-precision decimals, manipulating *regular expressions*, and *unit testing*.

Specifications cover some parts of the standard library, but most modules are not. They are specified by their code, internal documentation, and test suites. However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of November 2019, the Python Package Index (PyPI), the official repository for third-party Python software, contains over 200,000packages with a wide range of functionality, including:

- Automation
- Data analytics
- Databases
- Documentation
- Graphical user interfaces
- Image processing
- Machine learning and Deep Learning

- Artificial Intelligence
- Mobile Applications
- Multimedia
- Networking
- Scientific computing
- System administration
- Test frameworks
- Text processing
- Web frameworks
- Web scraping

2.4.4 PYTHON DATA TYPES

TYPE	MUTABILITY	DESCRIPTION	SYNTAX
BOOL	IMMUTABLE	BOOLEAN VALUE	True False
COMPLEX	IMMUTABLE	SEQUENCE OF BYTES	Bytes([120,105,150,160])
DICT	MUTABLE	An associative array (or dictionary) of key and value pairs; can contain mixed types (keys and values). Keys must be an able hash type	{'cat': lion, tiger, panther}
FLOAT	IMMUTABLE	Double precision floating point number.	1.414
INT	IMMUTABLE	Integer of unlimited magnitude	42
STR	IMMUTABLE	A character string: sequence of Unicode codepoints	"" senatorial districts""

2.5 WHY SPEECH?

- It is the most natural form of human communication
- modern technology (telephones etc.), people can communicate over long distances:
  - IVR systems are virtually everywhere
  - Such automated systems can remain online 24/7
- Voice commands can free hands and eyes for other tasks:
  - Cars
  - Airplanes
  - Ships

2.6 REASONS FOR RAPID ADVANCES

- Improvements in recognition algorithms
- Availability of large corpus of training data
- Unprecedented growth in computation and memory
  - MHz to GHz CPUs, MB to GB memory

2.7 TECHNIQUES APPLIED

Two methods can be applied, but for this study, the web-based API is to be used.

2.7.1 ONLINE AND OFFLINE APPLICATION

This Web Speech API was introduced at the end of 2012. It allows web developers to provide speech input and text-to-speech features in the web browser. The HTML5 Speech Recognition API enables JavaScript to access a browser's audio stream and convert it to text. Specifically, it defines a simplified subset of JavaScript API, working with speech recognition. This technique definition from (Mozilla Developer, March 2019).

The offline speech recognition data is a voice recognition system with the ability to recognize the voice and deliver data from it. ***The system works offline***, meaning you do not need an internet connection to use it. The system is a Google creation found in every smartphone using Android as the operating system.

2.7.2 FEATURES OF AN EFFECTIVE ASR

Many speech recognition applications and devices are available, but the more advanced solutions use ***AI and machine learning***. They integrate grammar, syntax, structure, and composition of audio and voice signals to understand and process human speech. Ideally, they learn as they go — evolving responses with each interaction.

The best system also allows organizations to customize and adapt the technology to their specific requirements — everything from language and nuances of speech to brand recognition. For example:

- **Language weighting:** Improve precision by weighting specific words spoken frequently (such as product names or industry jargon) beyond terms already in the base vocabulary.
- **Speaker labeling:** Output a transcription that cites or tags each speaker's contributions to a multi-participant conversation.
- **Acoustics training:** Attend to the acoustical side of the business. Train the system to adapt to an acoustic environment (like the ambient noise in a call center) and speaker styles (like voice pitch, volume, and pace).
- **Profanity filtering:** Use filters to identify specific words or phrases and sanitize speech output.

2.8 COMPARISON OF EXISTING SYSTEMS

There are two types of speech recognition. One is called *speaker-dependent*, and the other is *speaker-independent*. Speaker–dependent software is commonly used for dictation software, while speaker–independent software is more common in telephone applications.

Speaker–dependent software works by learning the unique characteristics of a single person's voice similar to voice recognition. New users must first "train" the software by speaking to it, so the computer can analyze how the person talks. This often means users have to read a few pages of text to the computer before using the speech recognition software.

Speaker–independent software is designed to recognize anyone's voice, so no training is involved. This means it is the only real option for applications such as interactive voice response systems — where businesses can't ask callers to read pages of text before using the system. The downside is that speaker-independent software is generally less accurate than speaker-dependent software.

Generally, all speech recognition engines that are speaker-independent deal with this fact by limiting the grammars they use. Using a smaller list of recognized words makes the speech engine more likely to identify what a speaker says correctly.

This makes speaker–independent software ideal for most IVR systems and any application where many people will be using the same system. Speaker-dependent software is used more widely in dictation software, where only one person will use the system, and there is a need for comprehensive grammar.

The LumenVox Speech Engine, which powers all of our speech software, is speaker-independent. It is not dictation software, it is not the same as voice recognition, and it cannot recognize an unlimited number of words at once. It is designed for identifying specific information, primarily by callers, into a telephone IVR. It works well as a call router, auto–attendant, or any other application where designers know what sort of words a speaker is likely to say.

We take hundreds of hours of transcribed audio and use it to create a language model to build it. This database tells our Speech Engine what sounds represent mathematically — math is the only language computers recognize.

Because the audio we use to build the models contains hundreds of speakers, the Engine has a wide variety of voices it can recognize. This is what makes it speaker-independent.

When the Engine receives input from a speech application, it converts the speaker's audio into a mathematical representation and compares it to its internal models. This gives the Engine an idea of which sounds make up the audio, and it resembles those sounds to the words specified in the speech application's grammar.

This is not an exact process. Because there are many subtle variations on how words are pronounced, the Speech Engine can never be optimistic about what the speaker says. For instance, even humans can never know what somebody said to them if the audio is unclear. Consider how difficult it is to distinguish between the letters "t" and "b" when spelling a word.

Our speech recognition software handles this uncertainty by using a method based on probabilities. In the same way, a public opinion poll has a margin of error for a specific confidence threshold. The Speech Engine returns a *confidence score* for any audio it attempts to recognize. This score represents how likely it is that the Engine's recognition result matches what the speaker said.

2.9 CONTRIBUTION OF PROPOSED SYSTEM

- **In the educational sector**, students who are blind or have impaired vision can benefit from using this technology to convey words and then hear the computer recite them and use a computer by commanding it with their voice, instead of looking at it the screen or keyboard.
- **Tech companies: (automobiles)** simple voice commands can be used to initiate phone calls, select radio stations, play music from a compatible smartphone, MP3 player, or flash drive.
  Voice recognition capabilities vary between car make and model. Some of the most recent models offer Natural Language speech recognition in place of a fixed set of commands, allowing the driver to use complete sentences and common phrases without memorizing a set of fixed command words.
- **Military: In fighter air crafts**, Speech recognition is used for a wide range of cockpit functions. Voice commands are confirmed by visual and aural feedback. The system reduces the pilot workload and even allows the pilot to assign targets to his aircraft with simple voice commands.
- **Medical sectors:** The model can be implemented in the front-end or back-end of the medical documentation process. A large part of the clinician's interaction with the **EHR** (Electronic Health Record) involves navigation through the user interface using menus and tab/button clicks, which heavily depend on the keyboard and mouse. **Thus, voice-based navigation provides only modest ergonomic benefits.**

Finally, users can access the product through cloud API's or on IoT devices.

CHAPTER THREE

METHODOLOGY

3.0 INTRODUCTION

This chapter focuses on how to build and manage an AI product from the startup till its deployment.

Various methods involved in this process to mitigate biases in the AI Product (Speech Recognition for African English) will also be explained.

3.1 TRAINING AND EVALUATION

This subsection discusses **ML** and **Automated ML**. Diving deeper into ML and understanding the concepts behind ML models would help us understand its crucial role in the development phase of this product.

How models are trained to carry out tasks by observing historical data and tuning parameters to do this with high accuracy, understanding the mechanics of ML, model architecture, training data, and model evaluation are all critical elements of ML and are instrumental for successful **AI PRODUCT *MANAGEMENT*** (in this case African SRS).

When using AI in products, numerous uncertainties can alter the course of development, and by understanding the core concepts, we can stay ahead of the unknown and plan development accordingly.

Note: When building AI products, we likely interact with experts in the field, and understanding some of the lingoes will allow for effectiveness when managing these products (Keren Bajupe; Senior HCI developer at Figure 8).

Concepts :

- Overview of modeling
- Training data
- Model evaluation
- Transfer learning and Automated ML

3.1.1 OVERVIEW OF MODELLING

Modeling consists of constructing the core components that will serve as a framework for an **ML model** NN (Neural Networks) are some of the most common models used in ML.

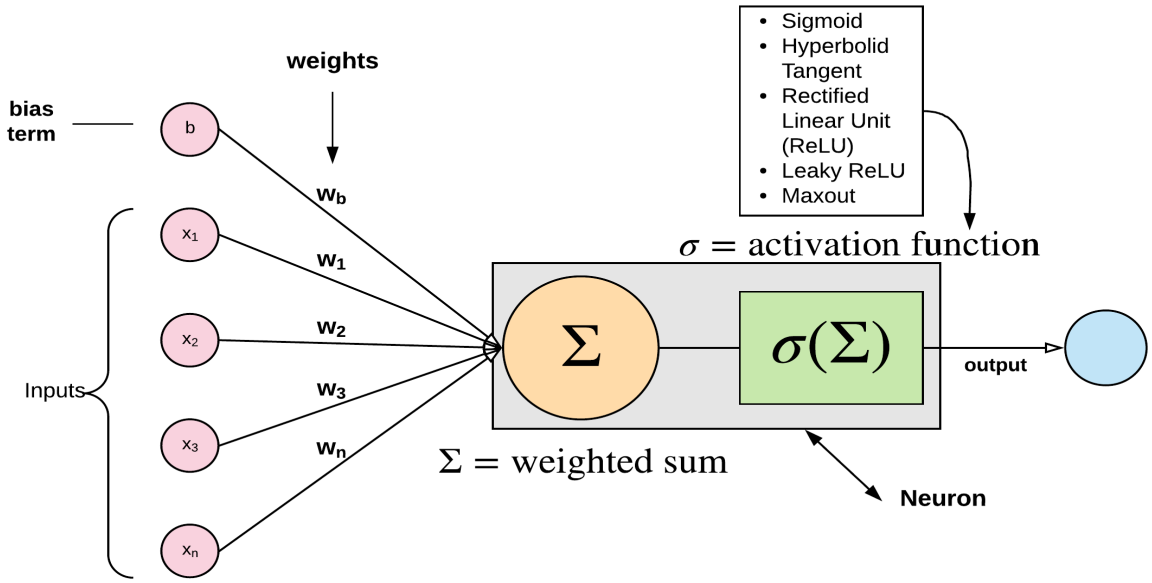
Neural Networks:

- First developed in 1950
- Series of computational layers
- Specialized nodes perform various computations

- Generally supervised ML technique to train model
- Predetermined network architecture

Note: layers are counted after the input layer (IL—L1—L2—OUTL).

3.1.2 ACTIVATION FUNCTION



DOES IT UNDERSTAND THE SPEAKER?

Consider a neural network with three inputs and the following weights:

- Open debug window x6
- Where do I live? x2
- What is my name? x2

Threshold = 7

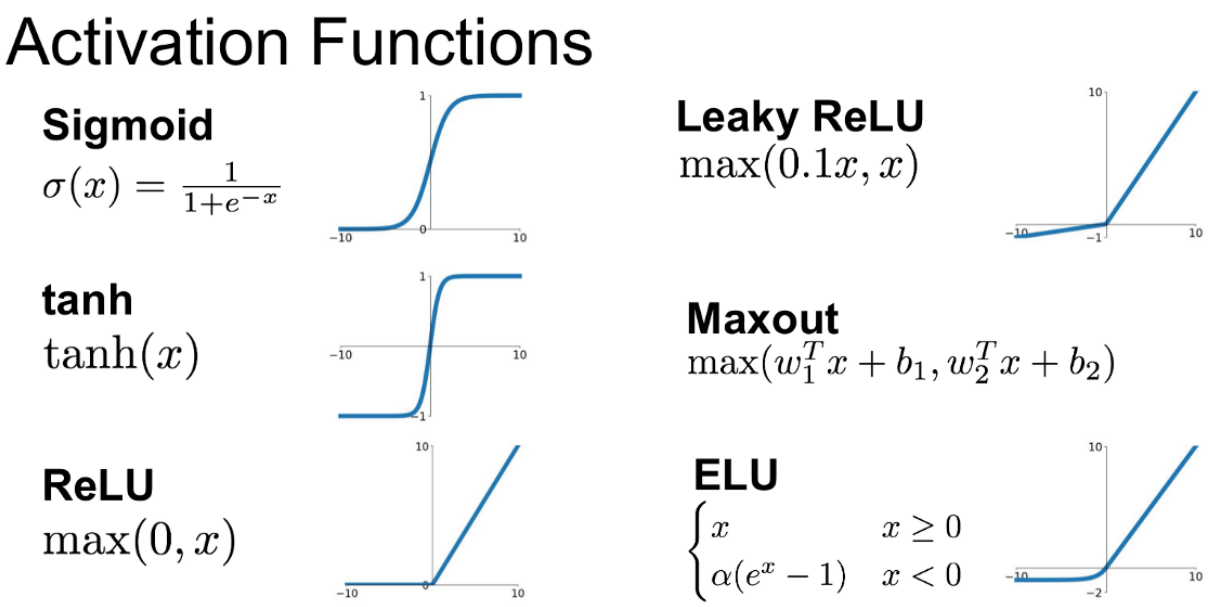
1x6+1x2+0x2 = 8

8 > 7 so YES;

A threshold above four will handle the case that the system understands "open debug window."

Handling large amounts of data such as large text documents, pixel images, or speech, the threshold would not be easy to determine where the activation function is applied.

Activation Functions are functions that we can use as decision boundaries to tell a Neural Network to OR not to send a signal to subsequent layers in the network, allowing us to pass continuous values instead of strictly 0 or 1 as it is essential for data flow in the model.



Model training will update weights to find the optimal parameters that handle all cases through backpropagation.

3.1.3 TRAINING DATA

The machine learns by looking at many examples of a given object and will eventually learn that a pattern in data can be associated with specific classification.

Machines unlike children are much more difficult to train and require special attention to ensure that it is learning the correct concept.

Data in ML

Learn by example

- Data defines model behaviour and performance
- Model parameters are updated based on training data
- A model will not learn if it is not in the data

Therefore, letting a model learn human voices; provide samples of human voices.

BAD data = **BAD** model

Giving our model a parrot talking, it would not be able to recognise human voices since it has been trained with a talking parrot voice.

NOTE: a model if not given a particular type of data it would never be able to infer on other types of data only after been trained on a given type or class of data would the model begin to make predictions of that type.

Training data is the key

- Models will only learn about data they are trained with
- Ensure that the data used to train model reflects real world data
- Use a diverse set of data to build a robust model

The right diversity in data

- Data that encompasses all likely scenario
- Equal amounts of the different types of data
- Correctly labelled data

Common issues with training data

- Unbalanced or biased data
- Data that does not reflect real world data
- Mislabeled data
- Insufficient data

Unbalanced data will cause a model to skew towards a particular outcome.

3.1.4 MODEL EVALUATION

Understanding model performance

- ✓ Need robust metrics to know how the model will perform after deployment
- ✓ Knowing which metrics to measure will help guide model development
- ✓ Model performance will determine the overall success of **AI** products

Data Evaluation

The model is evaluated against a test set of data

How we test?

$$\textbf{Precision} = \frac{TP}{TP + FP} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad 3.1$$

$$\textbf{Recall} = \frac{TP}{TP + FN} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad 3.2$$

$$F1 = \frac{2 * precision * recall}{precision + recall} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad 3.3$$

$$accuracy = \frac{TP+TN}{TP + FN + TN + FP} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad 3.4$$

$$Specificity = \frac{TN}{TN + FP} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad 3.5$$

Model precision

When a model makes a prediction how likely is that production correct?

I would use **precision** as the primary metric. Here, I need to aim for high precision at a reasonable threshold.

A precision level > **80%** is what I should aim for.

Model recall

How good a model is at identifying actual occurrence of object in the data giving us the percentage of object recalled by the model.

3.1.5 TRANSFER LEARNING AND AUTOMATED ML

Transfer learning is when we use the knowledge with the weights from an pre-trained network and adapt it to a new use case, generally one that requires different classes. This allows us to develop models with significantly less data and fewer changes because the neural network can retain knowledge it learns from previously trained data and slightly tunes parameters to fit our needs.

In transfer learning we can leverage existing trained models to solve new problems.

Automated ML takes us one step further by allowing us to build robust and scalable models without much machine learning experience.

- This is where cloud service providers such as Google cloud, IBM Watson offer ML model creation services. The way they work is that you bring your labelled data to they platform and it'll automatically be trained and tunned to fit your use case.
- Allows for quick prototyping
- Much less hassles and complexities
- Automatically determine best architecture for data types

AUTOMATED ML	CUSTOM MODELLING
Easy to get started with	Complete customizability
Robust enterprise support	Unlimited use cases
Cheap for quick development	Full control over parameter tunning
Limited use cases	Expensive to get started
Difficult to extend	Requires ML expertise
Data is accessible to provider	Limited use of external support

3.2 INTRODUCTION TO BUSINESS IMPACT

This section focuses on how to measure business impact and scale the **AI product for an enhanced productivity (speech recognition system model).**

Subsections will discuss:

- Benefits and challenges of AI initiatives
- Define and measure success metrics
- A/B testing and versioning
- Monitor and mitigate bias
- Continuous learning
- Compliance and ethics
- Scale

3.2.1 CASE STUDIES AND CHALLENGES

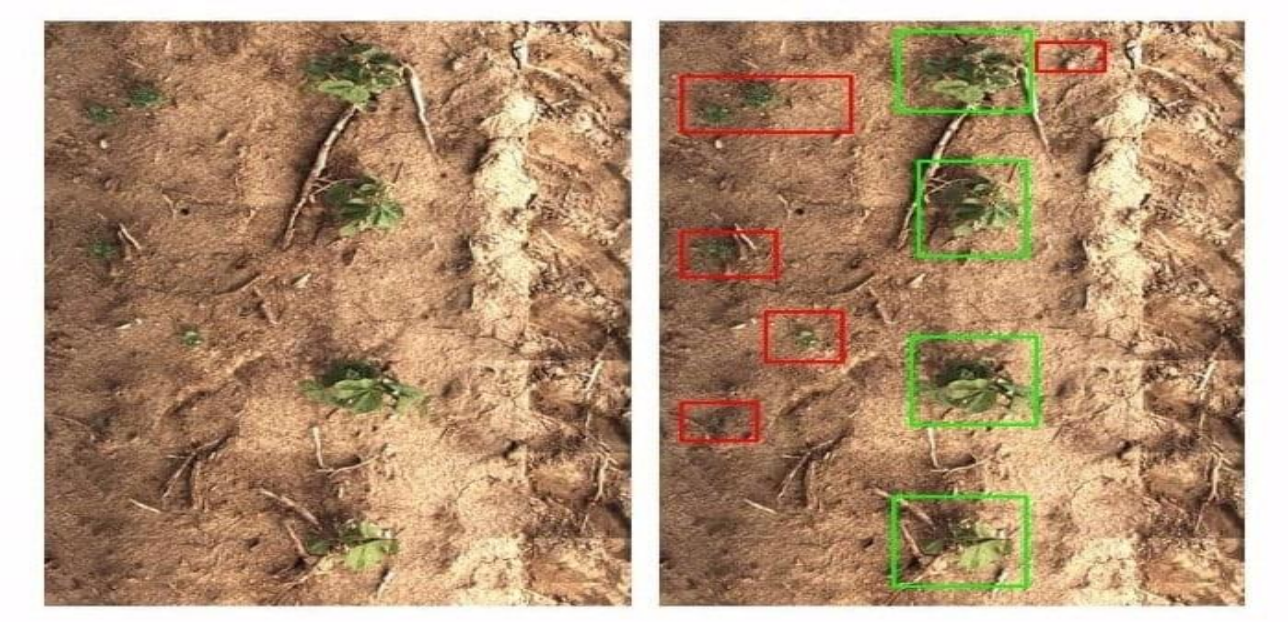
AIs leading benefits are enhanced products, processes, and better decisions.

For example, ***NETFLIX*** improves search results using AI which prevents user frustration and customer churn saving US$1 billion a year in potential loss revenue.

Blue river technology realised that farmers spend lot of money in pesticides the reason being that traditional farm equipment used are not precise.

It uses visual intelligence; instead of blanketing every square edge of the farm with pesticides the equipment uses computer vison algorithm to identify individual plants and precisely spraying only the weeds and not the crops.

The ***business value and benefits*** are tremendous in that, they have achieved **95%** accuracy, **50%** reduction in seed costs and **90%** reduction in herbicide spray.



3.2.2 MEASURING SUCCESS

When launching an **AI product** success metrics are used and before considering success metrics to be used the following are to be considered:

- ✓ Metrics should be easily measurable
- ✓ Metrics should correlate with the business problem
- ✓ Metrics should predict business outcome
- ✓ Metrics should be isolated to factors controlled by the group its measuring
- ✓ Metrics should be comparable to competitors metrics

58% have considered AI but only 12% have are actually using AI their business, to avoid falling into that trap we need to have a business goal.

Different industries have different business goals, once we have a business goal we then refine and revisit the success metrics such as:

- Customer experience
- Revenue gain
- Customer engagement
- Business process automation
- Better and faster decision making

Output vs outcome

AI products must be deployed to deliver specific and measurable business outcomes.

Outcome	Output
➢ Generate revenues ➢ Improve customer experience ➢ Increase user satisfaction ➢ Automate and save cost	➢ Accuracy ➢ Execution time ➢ Recall ➢ Precision

Output are success metrics to keep tracking and they should be given little attention unlike the business **outcome.**

3.2.3 A/B TESTING AND VERSIONING

A/B testing helps make more data driven decisions when it comes to evolving products and improving it.

It simply involves deploying two different models in production (**80%** controlled model, **20%** challenger model) and start tracking the performance metrics, decide which is the winning model and the decide whether to replace the first with the later or vice-versa.

One of the common mistake is declaring success too soon, so to avoid that consider the following:

Designing A/B test for models

- Deciding on a performance metric
- Deciding on test type based on performance metric
- Choosing a minimum effect size to be detected
- Determine the sample size
- Running the test until the sample size is reached

What else to consider:

- ✓ Cost benefit analysis:
  - is x% accuracy gain beneficial for business?
  - what if this slightly better model requires a much larger investment?
- ✓ Run test long enough to capture seasonality effects
- ✓ Control the experiment to avoid “novelty effects”- initial positive effects that wear off.

3.2.4 MONITOR BIAS

Bias creeping into a model can be bad for the ***business*** and also for user ***experience,*** hence monitoring and mitigating bias should be an ongoing initiative when launched and scaled.

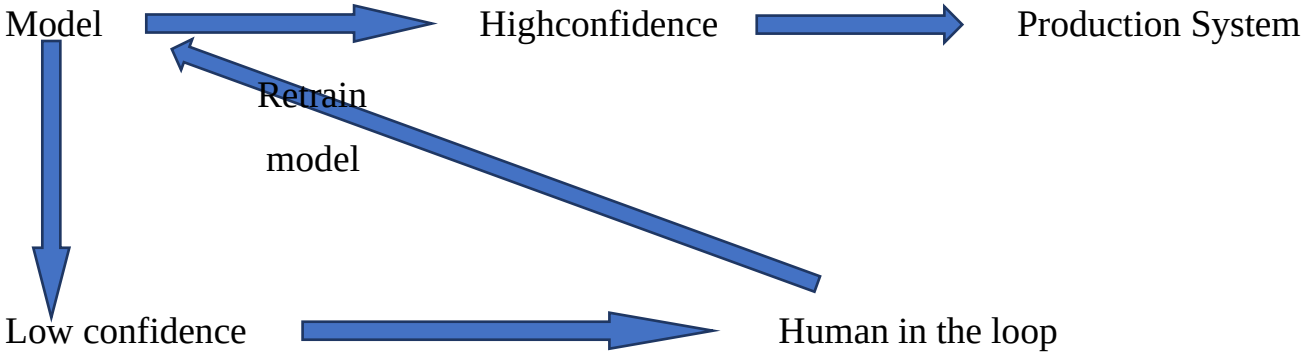
From a recent study, it is found that **ALEXA** and **GOOGLE** Assistant are **30%** less likely to understand non-American accents and even struggle more to understand women.

Recently, the ***Algorithmic Justice League's voice erasure project*** found that speech recognition systems from **Apple, Amazon, Google, IBM, and Microsoft** collectively achieve word error rates of 35% for African American voices versus 19% for White voices.
The crucial fact is that bias comes from training data and training data comes from humans hence to mitigate bias diversity is needed where subjective opinions matter.

How to solve unwanted bias?

- Awareness
  - ✓ Defining the decision our model is to solve for
  - ✓ Is the decision simple with white and black answers?
    Real world scenarios always have context
  - ✓ Is there a protected class or status involved in this decision?
- Data, Data, Data ……
  - ✓ Where is the data from? Could there be a sourcing bias?
  - ✓ Cover for enough examples and edge cases
  - ✓ Think about all end-users and test with them
  - ✓ Structure data gathering that allows for difference of opinions
    - Gather a diverse ML team that ask diverse questions
    - Annotate with diversity
- Iterate and learn
  - ✓ Be transparent on what data the system is trained on and its limitations
  - ✓ Be empathetic and understand your end-users will use the system differently. Build user experiences for failure/edge cases
  - ✓ Take feedback and have options to gracefully fall back
  - ✓ Ensure that the system is actively learning with examples, real world data with Human in the loop

3.2.5 CONTINUOUS LEARNING



Improving model accuracy and cost reduction with smart selection by:

1. Identifying the most valuable training data units for human annotation
2. Considering the four key criteria:
    - ✓ Low confidence
    - ✓ Uncertainty
    - ✓ Novelty
    - ✓ Class importance

3.2.6 COMPLIANCE AND ETHICS

AI products are raising serious concerns about:

- Trust
- Ethics
- Privacy
- Security
- Fairness

Building trust is the key; **41%** of voice assistant users have concerns about trust and privacy – ***Microsoft Report.***

The question arises: how do we balance benefits of AI technology with privacy?

A privacy first approach

- Data laws and AI coexist with privacy-first approach
- Deeper understanding and governance of data
- Understand the sensitivity of data (PII, PHI)
- Explain explicit consent and explain to customers how their data will be processed

Compliance and Certifications

As the AI product grows new industries will be targeted so making sure that the software is compliant to businesses in specific industries:

1. ISO information security management system
2. SOC type 2 certified
3. GDPR
4. HIPAA COMPLIANCE
5. pci security standards council
6. FedRAMP

3.2.7 SCALING A PRODUCT

After the previous stages, the AI product is scaled across companies as these approaches are considered:

1. Building a innovation centre with only pool of AI and Machine Learning experts then
2. span out to different units solving different problems using ML

the above consideration will need a lot of cross functional coordination and planning in order to the successful.

Please also note that these skills are useful no matter what type of AI product built.

CHAPTER FOUR

IMPLEMENTATION AND RESULTS

4.1 INTRODUCTION

This chapter shows all the work carried out in the project in software and hardware levels and the results of the implementation.

4.1.1 QUICK DESIGN

Any changes might occur during development according to system requirements. It is not a detailed design and includes only the important aspects of the system, which gives an idea of the system to the user. A quick design helps in developing the prototype. This system will develop as an application whose model is hosted on the cloud.

The prototyping model is applied in this development of the system as it is assumed that all the requirements may not be known at the start of the development of the system. It is usually used when a system does not exist or in case of a large and complex system where there is no manual process to determine the requirements.

This model allows the users to interact and experiment with a working model of the system known as prototype. The prototype gives the user an actual feel of the system. At any stage, if the user is not satisfied with the prototype, it can be discarded and an entirely new system can be developed.

4.1.2 MINIMUM VIABLE PRODUCT (MVP)

A **minimum viable product** (MVP) is a version of a product with just enough features to be usable by early customers who can then provide feedback for future product development.

A focus on releasing an MVP means that developers potentially avoid lengthy and (ultimately) unnecessary work. Instead, they iterate on working versions and respond to feedback, challenging and validating assumptions about a product's requirements. The term was coined and defined in 2001 by Frank Robinson and then popularized by Steve Blank and Eric Ries. It may also involve carrying out market analysis beforehand. The MVP is analogous to experimentation in the scientific method applied in the context of validating business hypotheses. It is utilized so that prospective entrepreneurs would know whether a given business idea would actually be viable and profitable by testing the assumptions behind a product or business idea.The concept can be used to validate a market need for a product and for incremental developments of an existing product. As it tests a potential business model to customers to see how the market would react, it is especially useful for new/startup companies who are more concerned with finding out where potential business opportunities exist rather than executing a prefabricated, isolated business model.

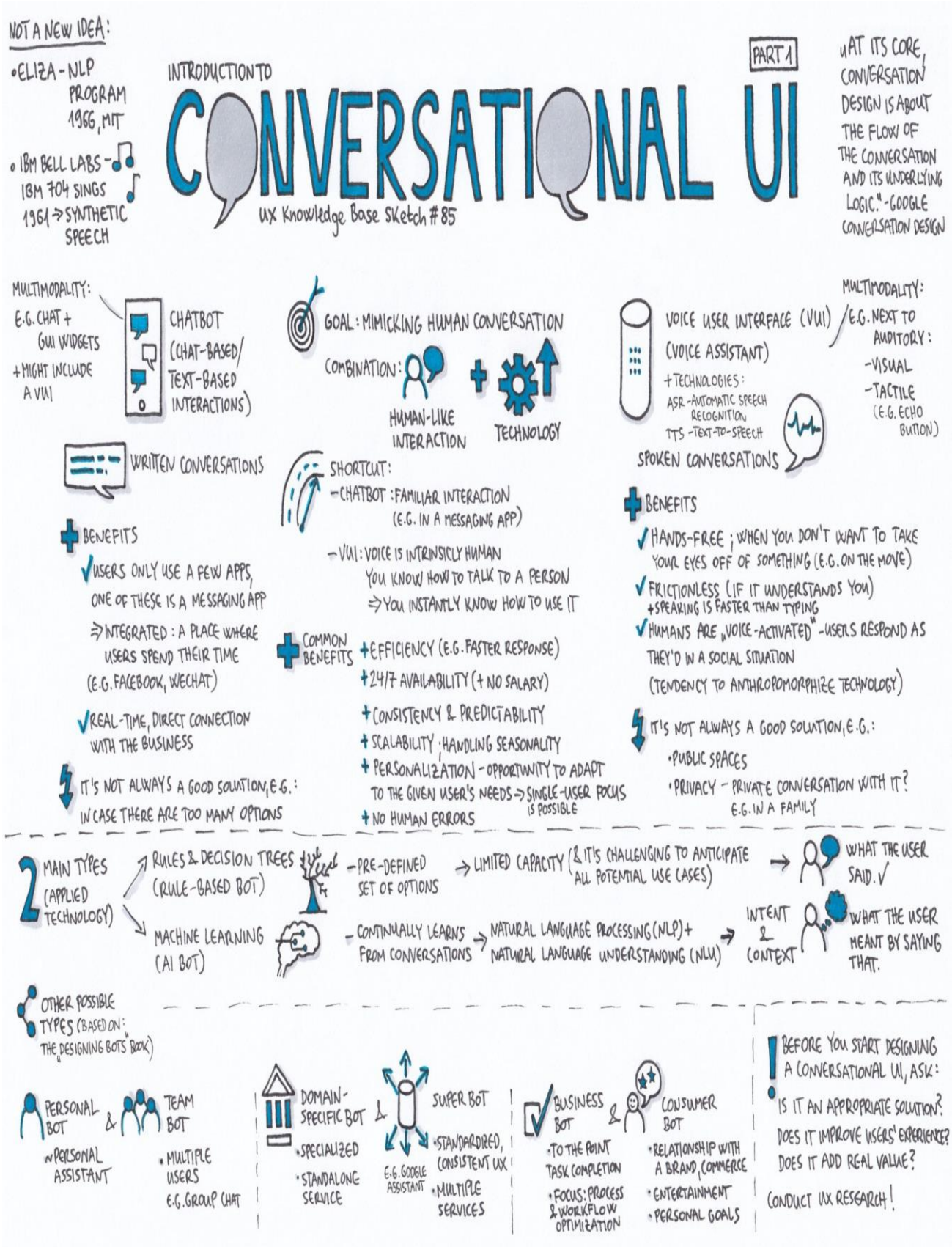
Purpose

- Be able to test a product hypothesis with minimal resource
- learning
- Reduce wasted engineering hours
- Get the product to early customers as soon as possible
- Base for other products
- To establish a builder's abilities in crafting the product required
- Speed up Brand building

The MVP will be a cross-platform speech recognition app and a REST-API. The app would first display a conversational user interface that collects voice data from the user and their country, adding it to the model's corpus of knowledge (pre-trained model).

The app is then able to recognize spoken words from anyone in any country. Furthermore, the REST API allows third-party applications to use the hosted online model for inference.

CONVERSATIONAL UI GUIDE



4.1.3 POST-MVP-DEPLOYMENT

I can improve the longevity of the product by:

- **Identifying customers needs and requirements:** Flexibility is required when balancing customers' needs and conditions concerning the price and performance of the product. **(Focusing on the outcome rather than output)**
- **Forming good communication skills:** The gap between market research and production should be minimum. The more layers of bureaucracy added between, the more likely it is that the design put forward by the research team and the final product will have several gaping holes.
- **Continuous learning of the AI model:** using a brilliant selection of training data to improve accuracy and **discarding** irrelevant data.

The training set won't know how to deal with values in a similar range because data distribution changes over time.

My product will learn from new data over time by displaying an interface that collects voice data from new users in any country and adds it to its corpus of knowledge. Once this is done, I can train the newly collected data and test it with its older data-set, improving the model's accuracy.

A/B testing as discussed in the previous chapter would prove helpful in improving my product, by letting me know what model performs better.

I can implement A/B testing by splitting the collected data into two versions (tested model and not yet tried model), and training the data to see how it performs with another newly collected data. This method further **reinforces** my model's confidence level to serve better when feeding with new data.

4.1.4 MONITORING BIAS

Steps taken to monitor/ mitigate unwanted bias are:

- **Narrowing the business problem**: I want to avoid solving too many scenarios, which would mean tons of labels across an unmanageable number of classes.
- **Structuring data gathering that allows for different opinions**.
- **Understanding the training data:** The more I know my data, the less likely I am to be surprised by objectionable labels.
- **Gathering a diverse ML team that asks different questions:** People from diverse backgrounds will inherently ask additional questions, which can help me catch problems before the model is in production.
- **Thinking about all end-users:** I can also avoid AI bias by anticipating how people will interact with the technology and what problems might arise.

- **Annotate with diversity:** An option is to source from a global crowd of annotators, providing different perspectives and supporting various languages, dialects, and geographically specific content.
- **and deploying with feedback in mind:** By opening a forum and discussion for feedback, I will ensure that my model maintains optimal performance levels for everyone.
- **a concrete plan to improve my model with the feedback:** Receiving feedback and giving my improved model feedback would constantly iterate my model towards higher accuracy.

4.2 RESULTS

The main personas includes:

- Students
- Tech Companies planning to use the product in their services
- Individuals
- Military

Epic-level Use cases:

1. **In the educational sector**, students who are blind or have impaired vision can benefit from using this technology to convey words and then hear the computer recite them and use a computer by commanding it with their voice, instead of looking at it the screen or keyboard.
2. **Companies: (automobiles)** simple voice commands can be used to initiate phone calls, select radio stations, play music from a compatible smartphone, MP3 player or flash drive. Voice recognition capabilities vary between car make and model. Some of the most recent models offer Natural Language speech recognition in place of a fixed set of commands, allowing the driver to use complete sentences and common phrases without memorizing a set of fixed command words.
3. In **fighter air crafts**, Speech recognition is used for a wide range of cockpit functions.Voice commands are confirmed by visual and/or aural feedback. The system reduces the pilot workload and even allow the pilot to assign targets to his aircraft with simple voice commands.
4. **Medical sectors:** The model can be implemented in the front-end or back-end of the medical documentation process. A large part of the clinician's interaction with the **EHR** (Electronic Health Record) involves navigation through the user interface using menus and tab/button clicks, which heavily depend on the keyboard and mouse. **Thus, voice-based navigation provides only modest ergonomic benefits.**

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 SUMMARY

"Product Managers play a critical role in making sure that the right product gets built. This starts by deeply understanding your users and the problems that you are solving for, then working with a number of different teams to identify and build a solution. After launching a product, it's an amazing feeling when you see people using your product in the real world." —**ALEX KING, PRODUCT MANAGER AT UBER**

Influencing the execution of product development as a Product Manager is one of the most highly-coveted roles in technology. **In This project,** I have highlighted **how to Envision** and lead the product strategy of industry-defining products, and learn to successfully bring it to the market.

5.2 CONCLUSION

The importance of the study of product management cannot be over emphasized in our society today as it helps to save a lot of situation concerning information confidentiality, integrity, and customer churn, and further helps to improve productivity in AI devices.

In this project i have presented a new method of approach for the implementation of AI products, how to improve productivity, and to maximize customer satisfaction in Africa .

5.3 RECOMMENDATIONS

Due to the undergraduate understanding of this deep learning project, the model needs room for improvement. The model's performance can, therefore, be further improved by implementing a Go-to-market plan:

- **Planning phase**(pre-launch): Securing partnerships with educational institutions and tech companies where the system can be tested, evaluated, and proven effective.
- **Launch phase**: When tested to be effective, the product can now be made available to other institutions and tech companies for data collection and analysis through university boards, social media pages of other tech companies.
- **Post-launch**: The product is on solid footing to succeed in the IT market through recommendations from supported users and a strong web presence to quickly deploy and close sales.

Also, injecting a new set of audio content into the model and re-training every month would increase the model's accuracy.

REFERENCES

My Udacity project (Bertelsmann AI product manager Nanodegree)

- GITHUB: AyeniTrust/Udacity-AI-Product-Manager-Projects (github.com)

Real Python: The Ultimate Guide To Speech Recognition With Python – Real Python

Hypothesis: EP1160767A2 - Speech recognition with contextual hypothesis probabilities - Google Patents

Speech Recognition: Speech recognition - Wikipedia

Implementation of SR Systems: Design and Implementation of Speech Recognition Systems (cmu.edu)

IBM CLOUD: Speech to Text - IBM Cloud API Docs

Speech Recognition: What is Speech Recognition? | IBM

Offline Speech Recognition Data - How to Stop Offline Speech Download? (ergonotes.com)
Types of Speech Recognition (lumenvox.com)

Towardsdatascience.com: 5 Types of Machine Learning Bias every Data Scientist should know

Thiang and SuryoWijoyo,(2011), "Speech Recognition Using Linear Predictive Coding and Artificial Neural Network for Controlling Movement of Mobile Robot," in Proceedings of International Conference on Information and Electronics Engineering (IPCSIT), Singapore, IACSIT Press, 179-183

Ms.Vimala.C and Dr.V.Radha,(2012), "Speaker Independent Isolated Speech Recognition System for Tamil Language using HMM," in Proceedings International Conference on Communication Technology and System Design 2011, Procedia Engineering, 1097 – 1102.

CiniKuriana, KannanBalakrishnan,(2011), "Development & evaluation of different acoustic models for Malayalam continuous speech recognition," in Proceedings of International Conference on Communication Technology and System Design 2011 Published by Elsevier Ltd, 1081-1088.

Suma Swamy,K.VRamakrishnan,(2013), “An Efficient Speech Recognition System”, Computer Science & Engineering: An InternationalJournal(CSEIJ), 21-27.

Annu Choudhary, Mr. R.S. Chauhan, Mr. Gautam Gupta,(2012), "Automatic Speech Recognition System for Isolated & Connected Words of Hindi Language By Using Hidden

Markov Model Toolkit (HTK)," in Proceedings of International Conference on Emerging Trends in Engineering and Technology,244– 252.

Preeti Saini, Parneet Kaur, MohitDua,(2013), "Hindi Automatic Speech Recognition Using HTK," International Journal of Engineering Trends and Technology (IJETT),"

Md. Akkas Ali, Manwar Hossain, Mohammad NuruzzamanBhuiyan,(2013), "Automatic Speech Recognition Technique for Bangla Words," International Journal of Advanced Science and Technology, 51-60.

Maya Moneykumar, Elizabeth Sherly, Win Sam Varghese,(2014), "Malayalam Word Identification for Speech Recognition System" An International Journal of Engineering Sciences, Special Issue Dravidian, 22-26

Jitendra Singh Pokhariya and Dr. Sanjay Mathur,(2014), "Sanskrit Speech Recognition using Hidden Markov Model Toolkit," International Journal of Engineering Research & Technology (IJERT), 93-98.

GeetaNijhawan and Dr. M.K Soni, (2014), "Real Time Speaker Recognition System for Hindi Words," International Journal of Information Engineering and Electronic Business, 35-40.

MVP: https://en.m.wikipedia.org/wiki/Minimum_viable_product

"What is a Minimum Viable Product (MVP)? - Definition from Techopedia".

Ries, Eric(August 3, 2009). "Minimum Viable Product: a guide".

"MVP: A maximally misunderstood idea". Slalom. Retrieved 2019-12-04.

"SyncDev methodology". SyncDev. Retrieved May 16, 2016.

W. S. Junk, "The Dynamic Balance Between Cost, Schedule, Features, and Quality in Software Development Projects", Computer Science Dept., University of Idaho, SEPM-001, April 2000.

Eric Ries, March 23, 2009, Venture Hacks interview: "What is the minimum viable product?", Lessons Learned

Perfection By Subtraction – The Minimum Feature Set

Holiday, Ryan The single worst marketing decision you can make The Next Web. 1 April 2015

"A Minimum Viable Product Is Not a Product, It's a Process: Building Product, Experimentation, MVP". YC Startup Library. Retrieved 2020-10-17.

"The Scientific Method for Startups: Building Product, Experimentation, KPI". YC Startup Library. Retrieved 2020-10-17.

Blank, Steve(2013-05-01). "Why the Lean Start-Up Changes Everything". Harvard Business Review (May 2013). ISSN 0017-8012. Retrieved 2020-10-17.