Project Report

On

Predictive Maintenance for an Industrial Application

Undertaken at

Gyrus AI, Bangalore, Karnataka

Submitted to

Department of Computer Science & Engineering, Tezpur University, Assam

Under the Guidance of

Mr. Abhishek Singh
Staff Engineer
Gyrus AI, Bangalore, Karnataka

&

Dr. ROSY SARMAH
Assistant Professor
Department of Computer Science & Engineering
Tezpur Central University, Tezpur, Assam

Submitted By

Mohan Kumar Sah (CSB16071)
(B.Tech, Computer Science & Engineering, Tezpur University)

Acknowledgement

There are many people who have added to the fruitful culmination of this venture and I seek this opportunity to offer my thanks to each and every one of them.

Above all else I thank the god-like to favor me with the open doors, quality, good faith and support. I thank my folks who have been a consistent wellspring of motivation and consolation.

With incredible joy and thankfulness, I augment my profound feeling of appreciation to **Mr. Chakra Parvathaneni**, Co-Founder and Director, Gyrus AI, Karnataka, Bangalore for giving me the chance to work as Intern at Gyrus AI.

I feel privileged to offer thanks and gratitude to **Mr. Nagaraj Krishnamurthy**, Director Software, Gyrus AI, Karnataka, Bangalore, **Mr. Abishek Singh**, Staff Engineer, Gyrus AI, Karnataka, Bangalore and **Dr. Rosy Sarmah**, Assistant Professor, Department of Computer Science & Engineering, Tezpur University, Assam for his help and guidance during Internship.

Ultimately I thank every one of those individuals who are directly or indirectly related with the finishing of this project.

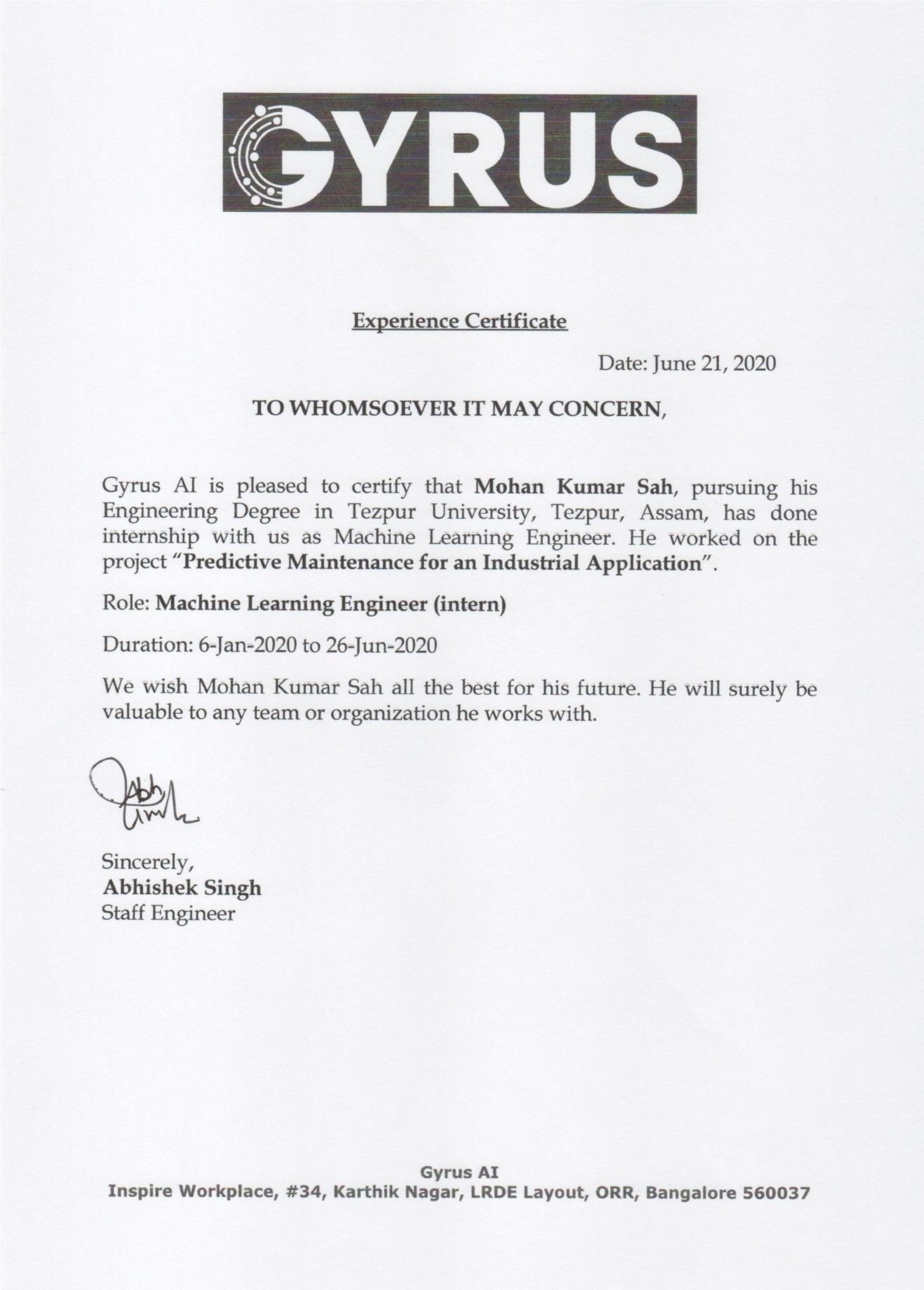
Declaration

I hereby declare that the project report on "***Predictive Maintenance for an Industrial Application***" is my original work. This written submission represents my ideas in my own words and wherever others ideas and words have been included, I have adequately cited and provided references to the original sources. I also declare that I have adhered to all the principles of the academic honesty and integrity and have not misrepresented or falsified any idea/data/fact/source in my submission.

Mohan Kumar Sah
20/6/2020

MOHAN KUMAR SAH
B.Tech (CSE dept. , 8th Sem.)
Roll No: CSB16071
Tezpur Central University, Tezpur, Assam

Internship Certificate



GYRUS

Experience Certificate

Date: June 21, 2020

TO WHOMSOEVER IT MAY CONCERN,

Gyrus AI is pleased to certify that **Mohan Kumar Sah**, pursuing his Engineering Degree in Tezpur University, Tezpur, Assam, has done internship with us as Machine Learning Engineer. He worked on the project **"Predictive Maintenance for an Industrial Application"**.

Role: **Machine Learning Engineer (intern)**

Duration: 6-Jan-2020 to 26-Jun-2020

We wish Mohan Kumar Sah all the best for his future. He will surely be valuable to any team or organization he works with.

Sincerely,
Abhishek Singh
Staff Engineer

Gyrus AI
Inspire Workplace, #34, Karthik Nagar, LRDE Layout, ORR, Bangalore 560037

Department of Computer Science and Engineering
TEZPUR UNIVERSITY
Tezpur-784028, Assam, India

CERTIFICATE

This is to certify that the report entitled ***"Predictive Maintenance for an Industrial Application"*** submitted to the **Department of Computer Science and Engineering**, Tezpur University in partial fulfilment for the award of the degree of Bachelor of Technology in **Department of Computer Science and Engineering**, is a record of project out by **MOHAN KUMAR SAH** under my supervision during the period from 6th January 2020 to 26th June 2020. All support received by him from various sources have been duly acknowledged. No part of this report has been submitted elsewhere.

Date:

Dr. Rosy Sarmah
Assistant Professor
Dept. of Computer Science & Engineering
Tezpur Central University, Tezpur, Assam

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

TEZPUR UNIVERSITY

Tezpur-784028, Assam, India

CERTIFICATE

This is to certify that the report entitled ***"Predictive Maintenance for an Industrial Application"*** is a bonafide record of project work carried out by **MOHAN KUMAR SAH** submitted in partial fulfilment for the award of the degree of Bachelor of Technology in Department of Computer Science and Engineering during the period from 6th January 2020 to 26th June 2020. He has carried out his project work under the supervision of **Dr. Rosy Sarmah**, Assistant professor**,** Department of Computer Science and Engineering, Tezpur University.

This approval does not necessarily endorse or accept every statement made, opinion expressed or conclusion drawn as recorded in the report. It only signifies the acceptance of this report for the purpose for which it is submitted.

Date:

Dr. Bhogeswar Borah
HOD & Professor
Dept. of Computer Science & Engineering
Tezpur Central University, Tezpur, Assam

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
TEZPUR UNIVERSITY

Tezpur-784028, Assam, India

CERTIFICATE BY THE EXAMINER

This is to certify that the report entitled "***Predictive Maintenance for an Industrial Application***" submitted by Mohan Kumar Sah (CSB16071) in partial fulfilment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering has been examined by me and is found satisfactory for the award of the degree.

This approval does not necessarily endorse or accept every statement made, opinion expressed or conclusion drawn as recorded in the report. It only signifies the acceptance of this report for the purpose for which it is submitted.

Internal Examiner	**External Examiner**
Date:	Date:
Place: Tezpur University	Place: Tezpur University

CONTENTS

SI No Chapter **Page No**

CHAPTER 1
Introduction

1.1. About Gyrus

Gyrus is one of the promising Software companies in private firms, which was founded in 2018, having headquartered in Santa Clara, CA, USA. Gyrus develops different products like Artificial Intelligence and Machine learning models, various Algorithms and Framework in the space of Business Intelligence, Internet of product analytics, video analytics.

Machine learning and Artificial Intelligence is soon about to turn over the culture of every industry intensely, however there are several restrictions in achieving the full potential. Gyrus with its journey and gained experience in this area is trying to make the transformation for the application of machine learning and artificial intelligence in business very soft, remunerating and great offerings. Gyrus built many core technologies to cover this transformation smoothly. The various limitations in the implementation of these technologies in different areas such as Data residing in multi silos, insufficient data, asynchronous data logging, annotation of data, generating poor quality datasets are required to proofing with specialised technique. All the machine learning and artificial intelligence algorithms are pointed to specific surroundings with matrices of accuracy, precision and recall depending upon the use case.

Reliability:
The various solutions for different projects designed by Gyrus are based on concrete foundations of knowledge and innovation and guarantee the best dependency.

Expertise:
Gyrus team is comprised of very professionals who apply their expertise to impart the industry with best services.

Quality:
At Gyrus, their priority is to develop and implement the best solution made available to their consumers that does not compromise with quality.

Experience:

Gyrus, with its expertise team having numerous years of experience in the field, always strives to provide the best service to its clients.

VISION OF GYRUS

Gyrus keeps a vision to enable various companies to operate with unparalleled productivity and efficiency by the application of human intelligence. Applying Artificial intelligence and Machine learning to implement predictive analysis, generate insight, and do monotonous activities and also to perform tasks at exceptional scale.

MISSION OF GYRUS

To make their customers delighted with best in segment technologies and support. Generate Ideas for new applications, use cases, Innovate and develop automation along with ideas. Be committed to their goals and product with best quality.

VALUES OF GYRUS

Gyrus is intended to remain:

- ✓ Focused to their customers
- ✓ Provide winner technologies in every segment
- ✓ Smartly provide the solution without creating much disturbance
- ✓ Honest by their dreams and commitment
- ✓ Modest and kind to the each comprising staff and customers

1.2. Project Profile

Title	Predictive Maintenance for an Industrial Application
Organization	Gyrus AI, Bangalore, Karnataka
Done By	Mohan Kumar Sah (CSB16071)
Role	Machine Learning Intern
Project Guide	**Mr. Abhishek Singh**, Staff Engineer at Gyrus AI, Bangalore
Internal Guide	**Dr. Rosy Sarmah**, Assistant Professor at Tezpur University, Assam
Duration	6 Months, 6th January, 2020 to 26th June, 2020

1.3. Problem Assigned

1. Develop an efficient TSR model on German Traffic Sign Recognition Benchmark (GTSRB) dataset to recognize the input traffic sign image and understand the concept of frame and neural network which have been used for Predictive Maintenance.

2. Get familiar with time series data, understand how to develop model on time series data and develop an efficient predictive model on Turbofan Engine Degradation Simulation dataset to

predict that which engine will need maintenance after 30 days.

Note: The Company wants an efficient model for Predictive Maintenance and for this my guide instructed to do experiment with multiple neural network architecture and select the best model based on accuracy and size.

1.4. Dataset

A **Data set** is a set or assortment of information. This set is normally presented in a tabular pattern.

For developing the machine learning models we divide the whole dataset into three parts for different purposes.

1. **Training Dataset:** This dataset will be utilized for training the model.
2. **Validation Dataset:** This dataset will be utilized for giving an impartial assessment of a model fit on the training dataset.
3. **Testing Dataset:** This dataset will be utilized for an unprejudiced assessment of the last model by computing the precision on this dataset.

I have worked on two datasets.

1. **German Traffic Sign Recognition Benchmark** Dataset for Traffic Sign Recognition [2].
2. **Turbofan Engine Degradation Simulation** Dataset for Predictive Maintenance [3].

German Traffic Sign Recognition Benchmark (GTSRB) Dataset

This dataset is an image dataset which consists of 43 traffic sign classes and 50,000 images. Each image represents a particular traffic sign. A sample of dataset with all the 43 classes is shown in Figure 1.2.1.

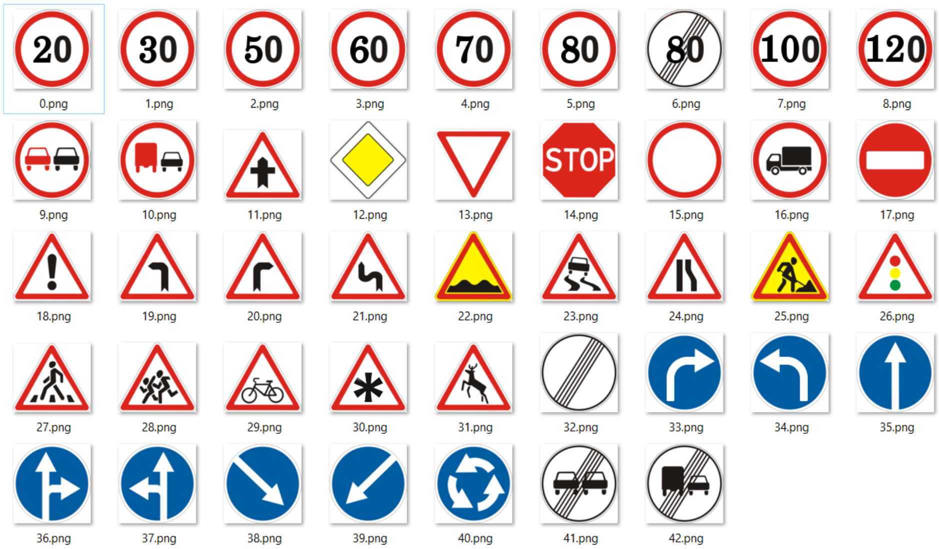


Figure 1.2.1: Sample of GTSRB dataset with all the classes

Turbofan Engine Degradation Simulation Dataset

This is a time series data from NASA. This dataset is normally represented in a tabular pattern where rows are the time steps and columns are variables.

This dataset consists of 33,727 rows, 27 variables (26 features and 1 target with 2 classes).

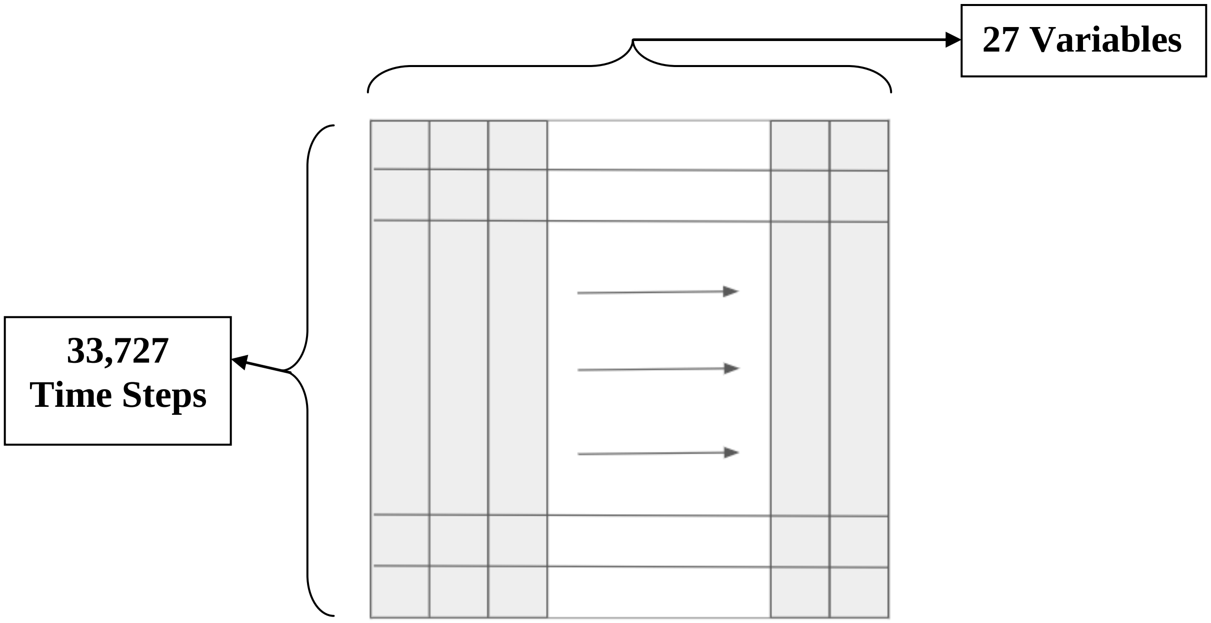


Figure 1.2.2: Dataset Structure

Rows are Time Steps and Columns are Variables (Features + Target). Each row belongs to a particular class (0 or 1), class "0" means the engine will not need any maintenance after 30 days and class "1" means the engine will need maintenance after 30 days.

CHAPTER 2
Initial System Study

2.1. Problem Definition

1. **Traffic Sign recognition**

   In current traffic management systems, there is a high likelihood that the driver may miss a portion of the traffic signs out and about due to congestion because of neighboring vehicles. With the constant development of vehicle numbers in urban agglomerations around the globe, this issue is just expected to deteriorate. To decrease the difficult Traffic Sign Recognition came into the scene.

   Traffic Sign recognition (TSR) is an innovation by which a vehicle can perceive the traffic signs put on the road. For example: Stop, One Way, No Entry, Speed Limit etc.

   Problem Statement

   Develop an efficient TSR model on German Traffic Sign Recognition Benchmark (GTSRB) dataset to recognize the input traffic sign image and understand the concept of frame and neural network which have been used for Predictive Maintenance.

   This model takes Image as Input and gives a class label for input image as output.

2. **Predictive Maintenance**

   There are a number of industrial machines that needs to keep running continuously. Such machines are rarely stopped. Hence regular maintenance of those machines is one of the most important tasks in any Factory. However, the process is quite tedious. Everyday records have to be taken of how the machines are performing. There is not just one machine but a number of them and a number of parameters to be checked. Most of these are done manually. If the machines are not regularly monitored there might raise failures which in turn would result in huge loss for the factory owners. To reduce the workload on the workers for regular monitoring of the factory machines Predictive Maintenance came into the scene.

   Predictive maintenance is the practice of determining the condition of equipment in order to estimate when maintenance should be performed - preventing not only catastrophic failures but also unnecessary maintenance, thus saving time and money.

   Problem Statement

   Get familiar with time series data, understand how to develop model on time series data and develop an efficient predictive model on Turbofan Engine Degradation Simulation dataset to predict that which engine will need maintenance after 30 days.

2.2. Proposed Solution

1. Solution for Traffic Sign Recognition

At Gyrus, I made an attempt to develop an efficient TSR model and my approach for this can be understood by points mentioned below.

1. Data Reading
    - Loading the data from a source and place it into the volatile memory for processing.
2. Data Preparation
    - Resizing all the images to a particular shape to get rid of the problem of image size because GTSRB dataset consists of having different shapes.
    - Splitting of the dataset into three parts i.e., training dataset, validation dataset and testing dataset.
3. Model Creation
    - Developing a CNN model which can be train and validate using training and validation dataset respectively.
4. Model Evaluation
    - Test the model by calculating the accuracy on testing dataset.
5. Model Prediction
    - Now the model is ready to make prediction on new data.

2. Solution for Predictive Maintenance

At Gyrus, I made an attempt to develop an efficient predictive model and my approach for this can be understood by points mentioned below.

1. Data Reading
    - Loading the data from a source and place it into the volatile memory for processing.
2. Data Preparation
    - Converting the tabular form of data into set of frames by considering the window size which can be shown by Figure 2.1.
        - Window size is the number of rows taken from tabular form of data to make a frame.
    - Splitting of the dataset into three parts i.e., training dataset, validation dataset and testing dataset.
3. Model Creation
    - Developing CNN, LSTM, CNN + LSTM and Convolutional LSTM models which can be train and validate using training and validation dataset respectively.
4. Model Evaluation
    - Testing all the models by calculating the accuracy on testing dataset.
5. Model Prediction

- Comparison of all the mentioned four models based on accuracy and size to select the best model for prediction on new data. Best model have high accuracy and less size.

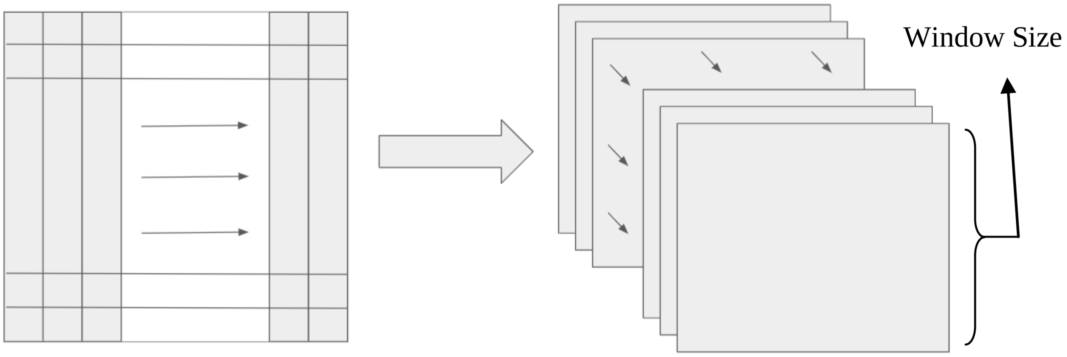


Figure 2.1: Converting tabular data to set of frames

Rows are Time Steps and Columns are Variables (Features + Target). Each frame belongs to a particular class (0 or 1), class “0” means the engine will not need any maintenance after 30 days and class “1” means the engine will need maintenance after 30 days.

CHAPTER 3
Requirements

3.1. Hardware

1. A system running **Windows 10** with admin privileges
2. CPU - Intel **2.4 GHz** Core i5
3. System Memory – Minimum **1GB RAM and 512 GB Space**

3.2. Language and Libraries

1. Python Programming Language

Python is an interpreted, high-level, general-purpose programming language. Python is one of the easiest programming languages because it uses English keywords. One of the best things about python is that it is open source even for the commercial applications which increases its popularity.

2. Libraries

1. **Numpy:** The full form of Numpy is Numerical Python. It is one the most basic library of python which can be used for working with arrays. It consists of functions which can help us when we are working in the domain of linear algebra, fourier transform and matrices.
2. **Pandas:** Pandas is an open source python library which provides efficient and powerful data structures for the python programming language which increases the popularity of python programming language. Pandas also provide data structure which helps us in working with "relational" or "labeled" data.
3. **Scikit-learn**: Scikit-learn is one the most notable open source python library which can be utilized for building up the machine learning algorithms. It consists of many simple and efficient tools which can be used for predictive data analysis. This library is very helpful when we will work on the problems related to classification, regression, clustering etc.
4. **Keras:** Keras is one of the most famous and ground-breaking open-source python library which can be utilized for machine learning and deep learning. Keras provides lots of functions related to layers of the neural networks, activation functions, loss functions, optimizers etc. Keras also provides functions which can help us in working with images and texts.

3.3. Software

1. Anaconda, Jupyter Notebook

Anaconda is a free and open-source dispersion of the Python and R programming languages for logical figuring, that aims to simplify package management and deployment. The dissemination incorporates data-science packages suitable for Windows, Linux, and macOS.

Jupyter Notebook is an open-source web application that permits you to make and offer records that contain live code, conditions, representations and account text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

CHAPTER 4
System Design

4.1. Algorithms with Architecture

1. Convolutional Neural Network (CNN)
2. Long Short Term Memory (LSTM)
3. CNN - LSTM
4. Convolutional Long Short Term Memory (ConvLSTM)

4.1.1. Convolutional Neural Networks (CNNs)

CNN stands for Convolutional Neural Networks. It is one of the most widely recognized neural networks which can be utilized for processing data which has a notable, grid-like topology. Example of such kind of data includes time series data and Image data [5].

A CNN can accept a picture as input, process it to appoint significance to all the objects present in that picture and separate the one object from other dependent on the significance allotted [6].

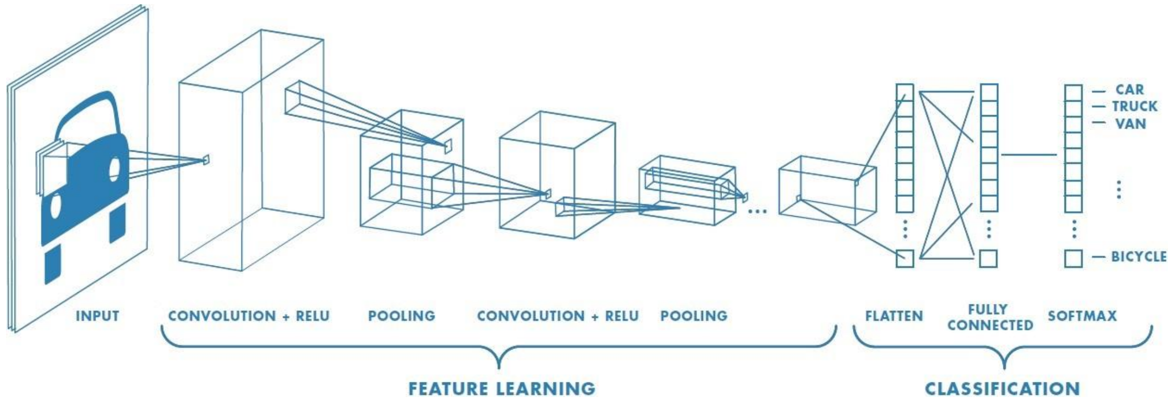


Figure 4.1: CNN Architecture

"*A basic CNN is a grouping of layers, and each layer of a CNN changes one volume of actuations to another through a differentiable limit*." In simple words we can say that each layer is liable for changing over the data from the qualities which are accessible in the past layers into some data which can be additionally passed to the following layers for additional speculation.

We can divide the CNN into two basic blocks [5]:

1. **Convolution Block**: This block essentially comprises of Convolution and Pooling Layer. The capacity of this block is to extract the features from the input image and pass to the fully connected block.
2. **Fully Connected Block**: This block basically comprises of a fully connected simple neural network architecture. The capacity of this block is to play out the errand of Classification dependent on the information got from the convolutional block.

Convolution Layer [7]

1. Convolution layer is the principal layer and it is responsible to extricate highlights from an input image.
2. By learning the image features, Convolution layer also preserves the relationship between pixels.

Mathematically an image can be represented as matrix which contains pixel values such matrix is called **Image Matrix**.

Filter is a small 2-D array that contains some values which determine the nature of the process.

Convolution is given by:

$$g(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x - s, y - t)$$

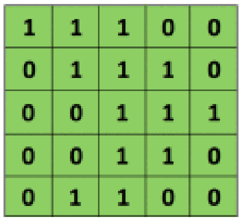
Where, ‘w’ is the input image of size M X N and ‘f’ is the filter of size m x n.

$$a = (m-1)/2 \qquad b = (n-1)/2$$

$$x = 0,1,2,\ldots, M-1 \qquad y = 0,1,2,\ldots, N-1$$

After convolution we will get ‘g’ of size (M - m + 1) X (N - n + 1) as output which is called Feature Map.

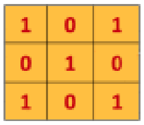
Next I present the procedure for convolution of an Image (I) with a Filter (F).



1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

5 x 5 Image, I

*



1	0	1
0	1	0
1	0	1

3 x 3 Filter, F

Steps for convolution of an Image using a filter

1. Slide the filter over the image.
2. Multiply each weight in the filter by the pixel of Image underneath.
3. Sum the individual products.

We have to repeat these three steps for getting a complete feature map.

A complete process for convolution of an Image (I) with a Filter (F) is shown below.

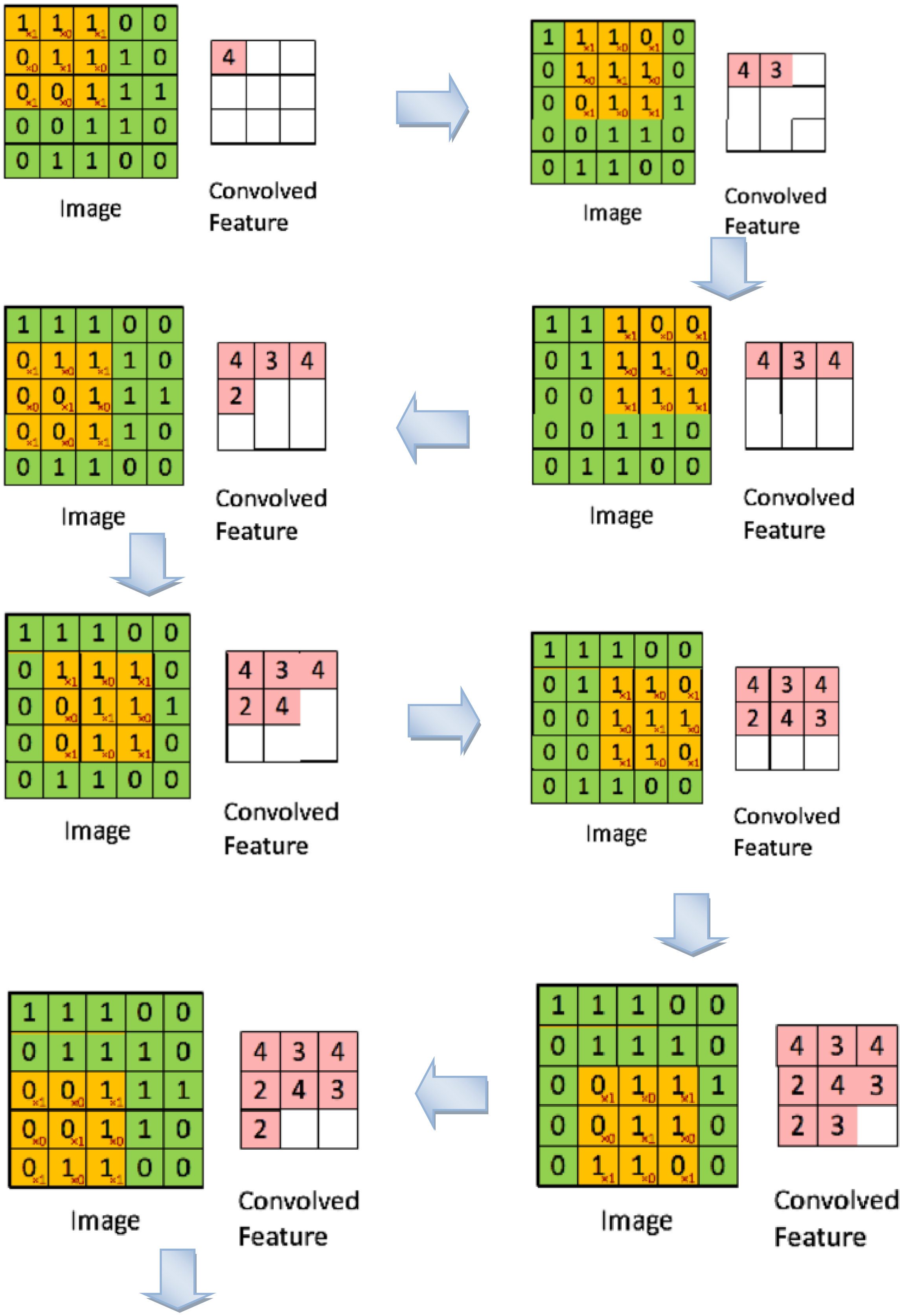
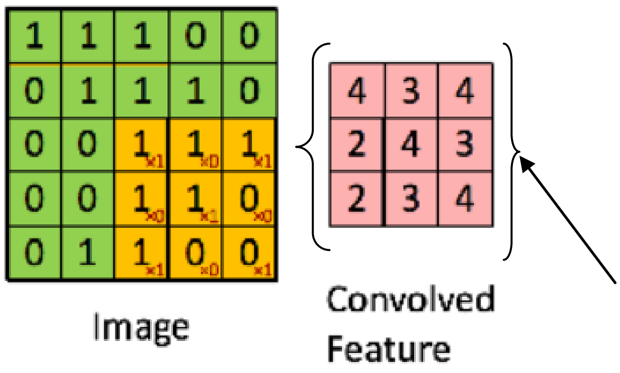


Image
Convolved Feature
Image
Convolved Feature
Image
Convolved Feature
Image
Convolved Feature
Image
Convolved Feature
Image
Convolved Feature
Image
Convolved Feature
Image
Convolved Feature



1	1	1	0	0
0	1	1	1	0
0	0	$1_{\times 1}$	$1_{\times 0}$	$1_{\times 1}$
0	0	$1_{\times 0}$	$1_{\times 1}$	$0_{\times 0}$
0	1	$1_{\times 1}$	$0_{\times 0}$	$0_{\times 1}$

4	3	4
2	4	3
2	3	4

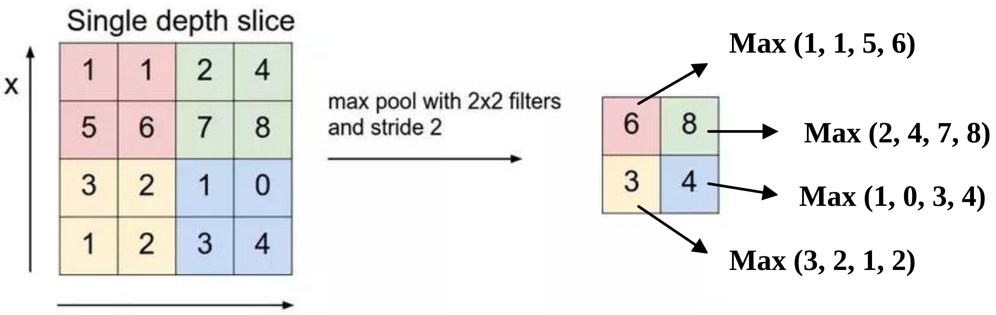
Output: Feature Map

Polling Layer [7]

1. Pooling layers are responsible for decreasing the number of parameters to consider the significant features only when the size of the input image is enormous.
2. Spatial pooling also called sub sampling or down sampling which helps in decreasing the number of parameters without affecting the significant data.

Different types spatial pooling:

1. **Max Pooling**: It returns the largest element from the feature map.
2. **Average Pooling**: It returns the average of all elements from the feature map.
3. **Sum Pooling**: It returns the sum of all elements from the feature map.



1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

6	8
3	4

Max (1, 1, 5, 6)

Max (2, 4, 7, 8)

Max (1, 0, 3, 4)

Max (3, 2, 1, 2)

Stride is the number of pixels shifts by filter over the input matrix.

Figure 4.2: Max Pooling

Fully Connected Layer [7]

The output of the polling layer is in the form of matrix which can flatten into vector and pass the resultant vector into a fully connected layer like a neural network. We also have an activation function in this layer which can be used to classify the output. We can understand this from the Figure 4.3.

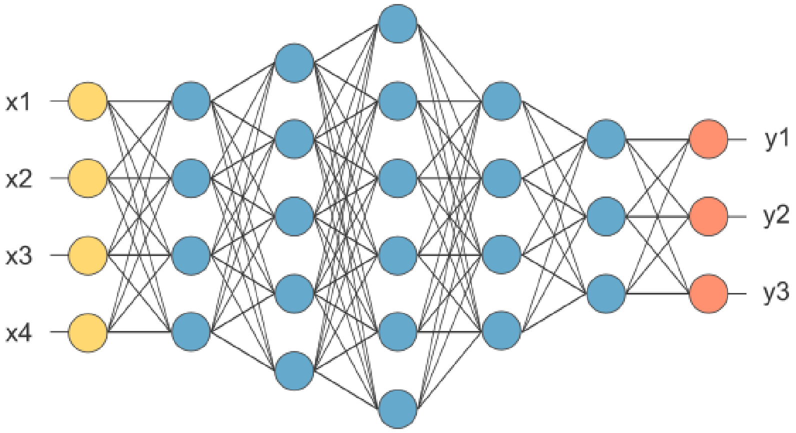


Figure 4.3: After pooling layer, flattened as FC layer

In the above Figure 4.3, the output matrix of the polling layer is converted into a vector (x1, x2, x3, x4 and so on) and passed into the fully connected layer and by using an activation function such as softmax or sigmoid the outputs are classified as (y1, y2, y3 and so on).

4.1.2. Long Short Term Memory (LSTM)

LSTM is a unique neural system which is accustomed to grouping, handling and making forecasts dependent on time arrangement information. One of the most significant highlights of LSTM is that it can recall the data for extensive stretch of time.

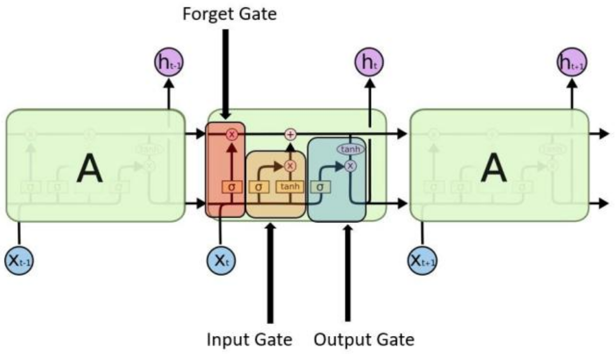


Figure 4.4: LSTM Architecture

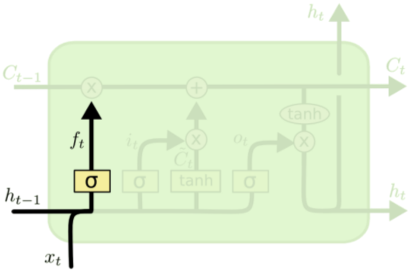
LSTM mainly consists of three gates which is shown in Figure 4.4

1. Forget gate
2. Input gate
3. Output gate

Forget Gate [8]

Forget Gate tells us about the past information that we have to remember.

At a specific time stamp, forget fate chooses which data will be discarded from the cell. Forget gate thinks about the past state (ht – 1) and current info (Xt) and gives a yield somewhere in the range of 0 and 1 for each number present in the cell state Ct - 1 by utilizing the sigmoid activation function.

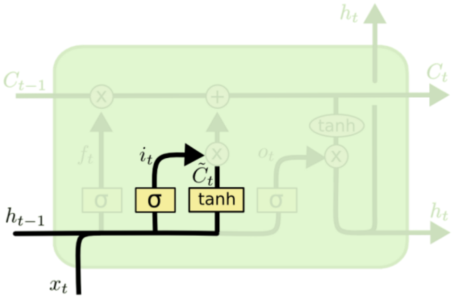


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Figure 4.5: Forget Gate

Update Gate/Input gate [8]

Update / Input gate tells us about the unit which is added to the current state.



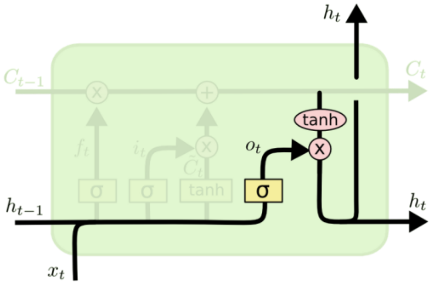
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Figure 4.6: Input Gate

Sigmoid function chooses which esteems to let through 0, 1 and tanh function offers weightage to the qualities which are passed choosing their degree of significance extending from-1 to 1**.**

Output Gate [8]

Output gate tells us about what portion of data from the current cell will present in the output.



$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Figure 4.7: Output Gate

Sigmoid function chooses which esteems to let through 0, 1 and tanh function offers weightage to the qualities which are passed choosing their degree of significance extending from-1 to 1 and increased with yield of Sigmoid.

4.1.3. CNN - LSTM

This is not new neural network architecture. It is only the mixture of Convolutional Neural Network and Long Short Term Memory. The architecture of this hybrid neural network is shown in Figure 4.8.

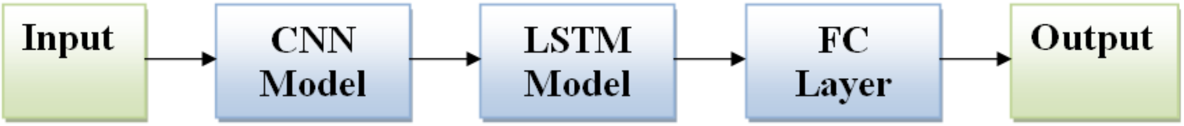


Figure 4.8: Generic Architecture of a CNN-LSTM Model

4.1.4. Convolutional LSTM (ConvLSTM)

ConvLSTM is just like the LSTM the only difference is that all the internal matrix multiplications of LSTM are exchanged with convolution operations.

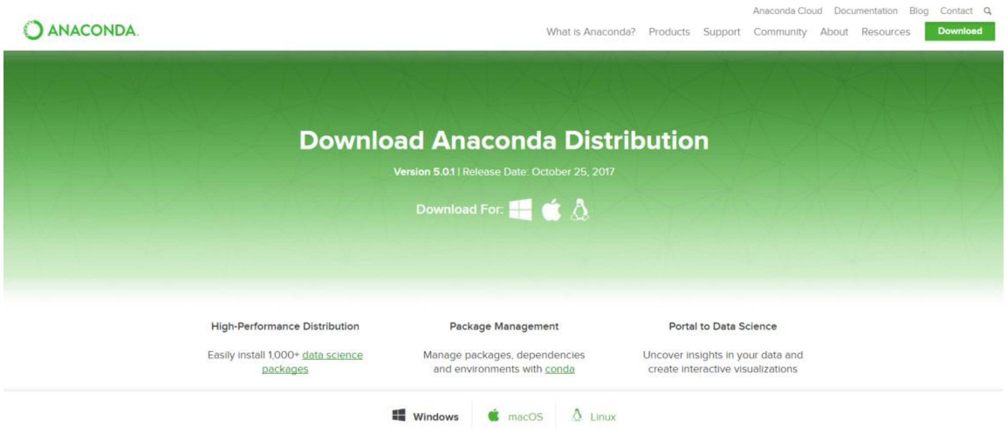
CHAPTER 5
System Implementation

5.1. System set-up required to develop the solution

For developing the solution we have to install the Software and all the Python libraries (mentioned in the section **3.3** and **3.2**) in our system.

5.2. Installation of Anaconda and Jupyter Notebook

Step1. Visit the Anaconda downloads page [12], link: Anaconda.com/downloads
After opening the link, the Page will look something like this:



Download Anaconda Distribution

Version 5.0.1 | Release Date: October 25, 2017

Download For:

High-Performance Distribution

Easily install 1,000+ data science packages

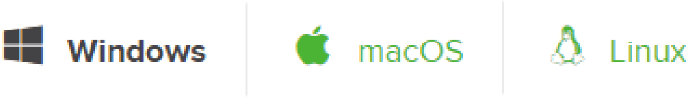
Package Management

Manage packages, dependencies and environments with conda

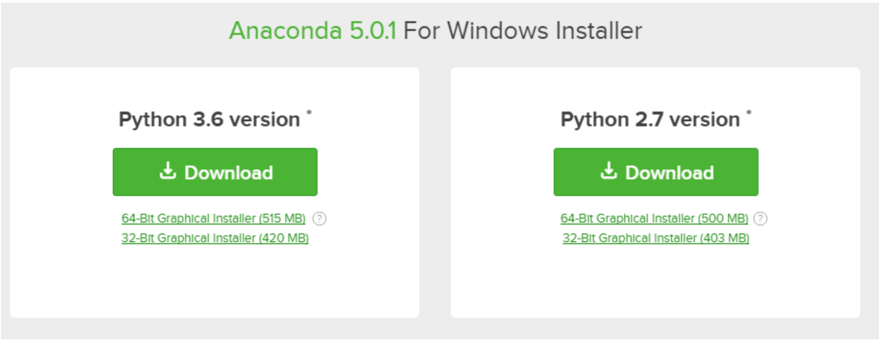
Portal to Data Science

Uncover insights in your data and create interactive visualizations

Step2. Select Windows from the three operating systems are listed.



Step3. Download
After selecting the operating system the page will look like this:


Anaconda 5.0.1 For Windows Installer

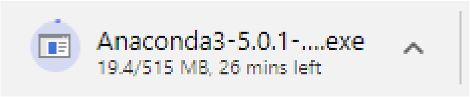
Python 3.6 version *

64-Bit Graphical Installer (515 MB)
32-Bit Graphical Installer (420 MB)

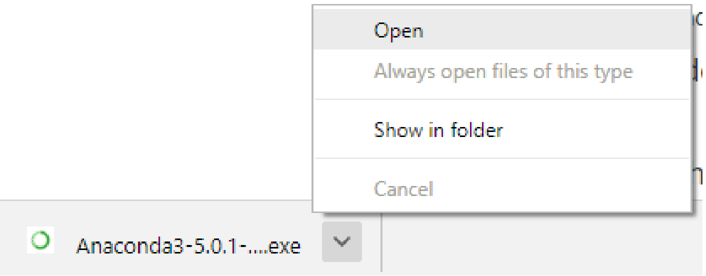
Python 2.7 version *

64-Bit Graphical Installer (500 MB)
32-Bit Graphical Installer (403 MB)

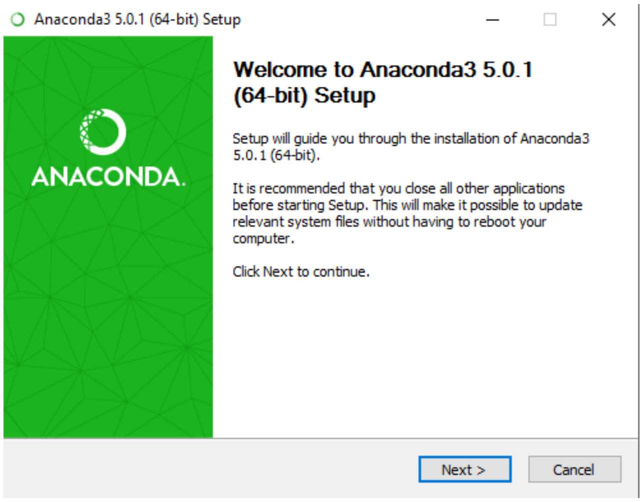
Select and download the Python 3.6 version. The downloading may take some time because the size is quite large so wait until the download finish.



Step4. Open and run the installer
Once the download finishes, open and run the ***.exe*** installer



After opening a "Welcome to Anacondas" screen will come. Confirm the installation by clicking the Next button.



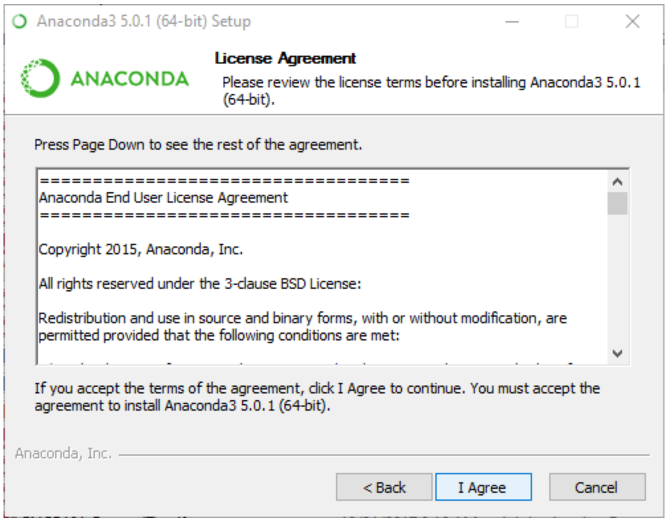
Welcome to Anaconda3 5.0.1 (64-bit) Setup

Setup will guide you through the installation of Anaconda3 5.0.1 (64-bit).

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

After clicking Next a “License Agreement” screen will come. Agree to the license by clicking the I Agree button.



License Agreement

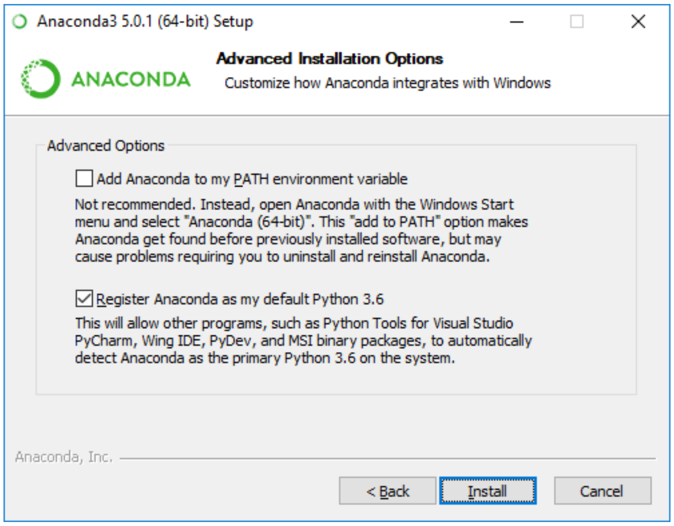
Please review the license terms before installing Anaconda3 5.0.1 (64-bit).

Press Page Down to see the rest of the agreement.

```
==================================
Anaconda End User License Agreement
==================================

Copyright 2015, Anaconda, Inc.

All rights reserved under the 3-clause BSD License:

Redistribution and use in source and binary forms, with or without modification, are
permitted provided that the following conditions are met:
```

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install Anaconda3 5.0.1 (64-bit).

After clicking on I Agree an “Advance Installation Options” screen will come. At the “Advanced Installation Options” screen, do not check "Add Anaconda to my PATH environment variable" however check “Register Anaconda as my default Python 3.6” and click on install button.



Advanced Installation Options

Customize how Anaconda integrates with Windows

Advanced Options

□ Add Anaconda to my PATH environment variable

Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requiring you to uninstall and reinstall Anaconda.

☑ Register Anaconda as my default Python 3.6

This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.6 on the system.

Step5. Open the Anaconda Prompt from the Windows search box.
Once the installation of Anaconda completes, you just go to the Windows search box, type Anaconda Prompt, select the Anaconda Prompt from the menu and open it [13].

Type the command “python –version” and press enter to verified Anaconda was installed.

```
> python --version
Python 3.6
```

For Starting Jupyter Notebook

1. Type the command Jupyter notebook and press enter to start Jupyter Notebook.

```
> jupyter notebook
```

5.3. Installation of Python Libraries

Command to install Numpy

```
> conda install -c anaconda numpy
```

Command to install Pandas

```
> conda install -c anaconda pandas
```

Command to install Scikitt-learn

```
> conda install -c anaconda scikit-learn
```

Command to install Keras

```
> conda install -c conda-forge keras
```

CHAPTER 6
Results

Results for Traffic Sign Recognition

- ✓ **Training Accuracy**: 98.91 %
- ✓ **Validation Accuracy**: 94.03 %
- ✓ **Testing Accuracy**: 96.04 %

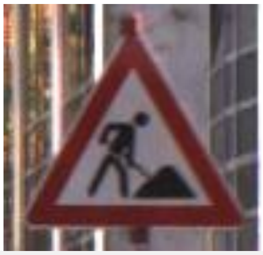
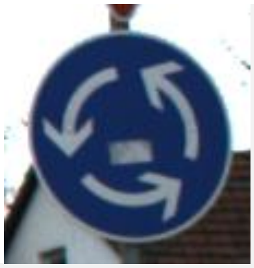
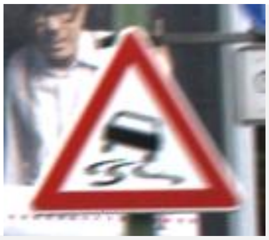
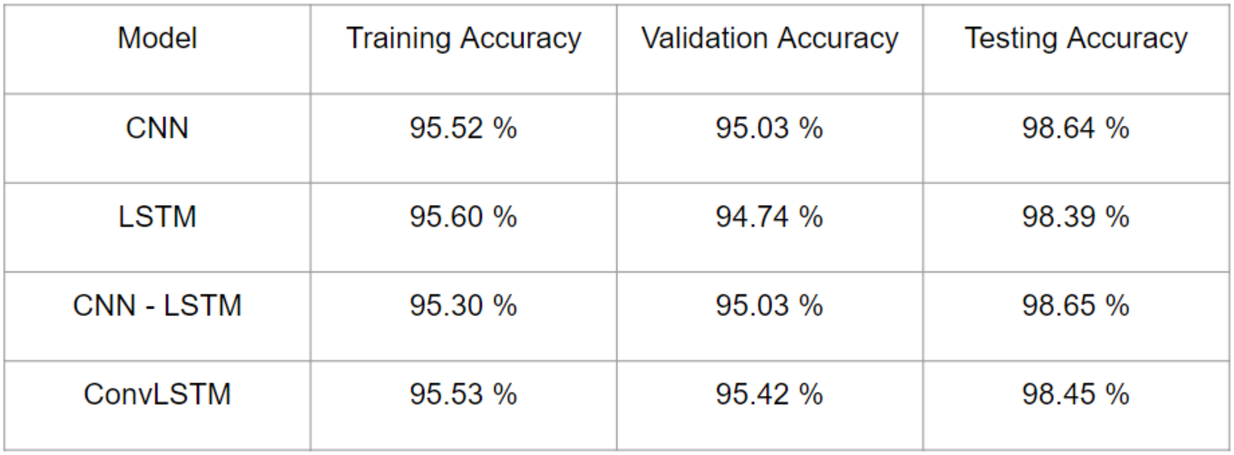
Input	Actual Class	Predicted Class / Output
	18	18
	25	25
	40	40
	23	23

Figure 6.1: Left side, middle and right side of all the four rows shown above are Input Image, Actual Class and Predicted Class or Output respectively.

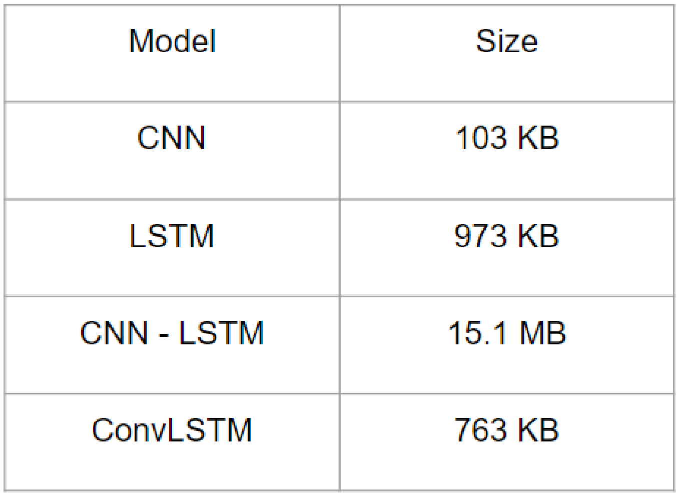
Results for Predictive Maintenance



Model	Training Accuracy	Validation Accuracy	Testing Accuracy
CNN	95.52 %	95.03 %	98.64 %
LSTM	95.60 %	94.74 %	98.39 %
CNN - LSTM	95.30 %	95.03 %	98.65 %
ConvLSTM	95.53 %	95.42 %	98.45 %

Table 6.2: Table of Comparison for Models based on Accuracy

After analyzing Table 6.2, we can conclude that we cannot select mode based on accuracy because all models are showing almost similar accuracy. So now we have to analyze models based on size to select the best model.



Model	Size
CNN	103 KB
LSTM	973 KB
CNN - LSTM	15.1 MB
ConvLSTM	763 KB

Table 6.3: Table of Comparison for Models based on Size

After analyzing Table 6.3, we can conclude that CNN model is the best model because it has less size.

After analyzing Table 6.2 and 6.3, we can conclude that CNN model is the best model because it has high accuracy as well as less size.

CHAPTER 7
Voice Chatbot

7.1. Introduction

I have completed all the assigned work a few days before ending the internship so the company instructed to develop a simple voice chatbot which will take voice as Input and gives voice as output.

Chatbot

A chatbot is a kind of conversational specialist, a PC program intended to mimic a canny discussion with at least one human user in normal language by means of sound-related or literary strategies.

Working of a Chatbot

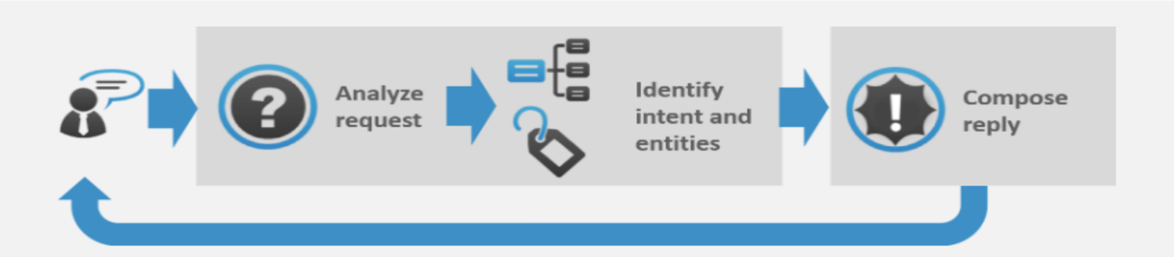


Figure 7.1.1: Working of a Chatbot

The working of a chatbot can be divided into two steps:

1. User request analysis
    - Dissects the solicitation of the client to distinguish the client's goal and to separate pertinent elements.
2. Returning the response
    - When the target of the client has been distinguished, the chatbot is dependable to give the most fitting reaction to the solicitation.

What is a Voice chatbot?

It is similar to the chatbot only difference is that it takes voice as input and also gives voice as output.

What is Chatbot Corpus ?

This is a corpus of dialog data that is included in the chatterbot module.

7.2. Logic used to build the Voice Chatbot

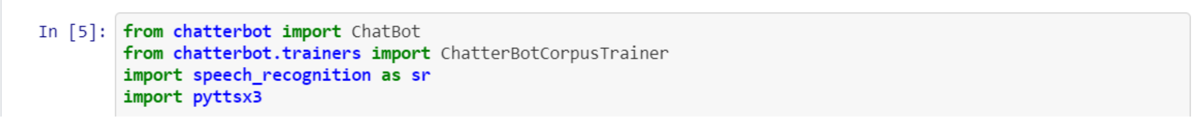
Logic to build the Voice Chatbot mainly consists of three steps:
First we have to develop a chatbot which can take the text as input and gives (reply) with text as output which can be used in step 2.

1. Converting the speech (user voice) into text
   - ✓ Recognize the voice of the user which is the input of the Voice Chatbot and convert the voice into string (text) for the next step.
2. Feed the text to chatbot which will give a reply to that text
3. Converting the reply into speech (chatbot voice)
   - ✓ Convert the text again to voice which is the output of the Voice Chatbot.

In python code I put these three steps in a infinite while loop and set a condition that if user say “bye” then loop breaks and program terminates.

7.3. Implementation and Results

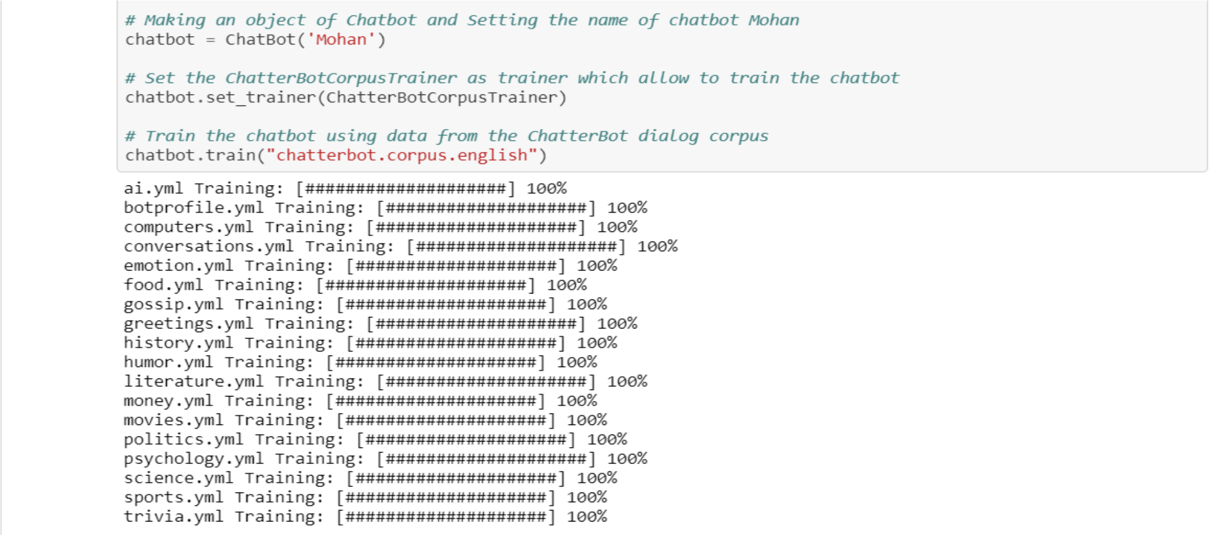
Importing necessary packages



```
In [5]: from chatterbot import ChatBot
        from chatterbot.trainers import ChatterBotCorpusTrainer
        import speech_recognition as sr
        import pyttsx3
```

Figure 7.3.1: Necessary Packages used for developing Voice Chatbot

Chatbot which can take the text and reply with text

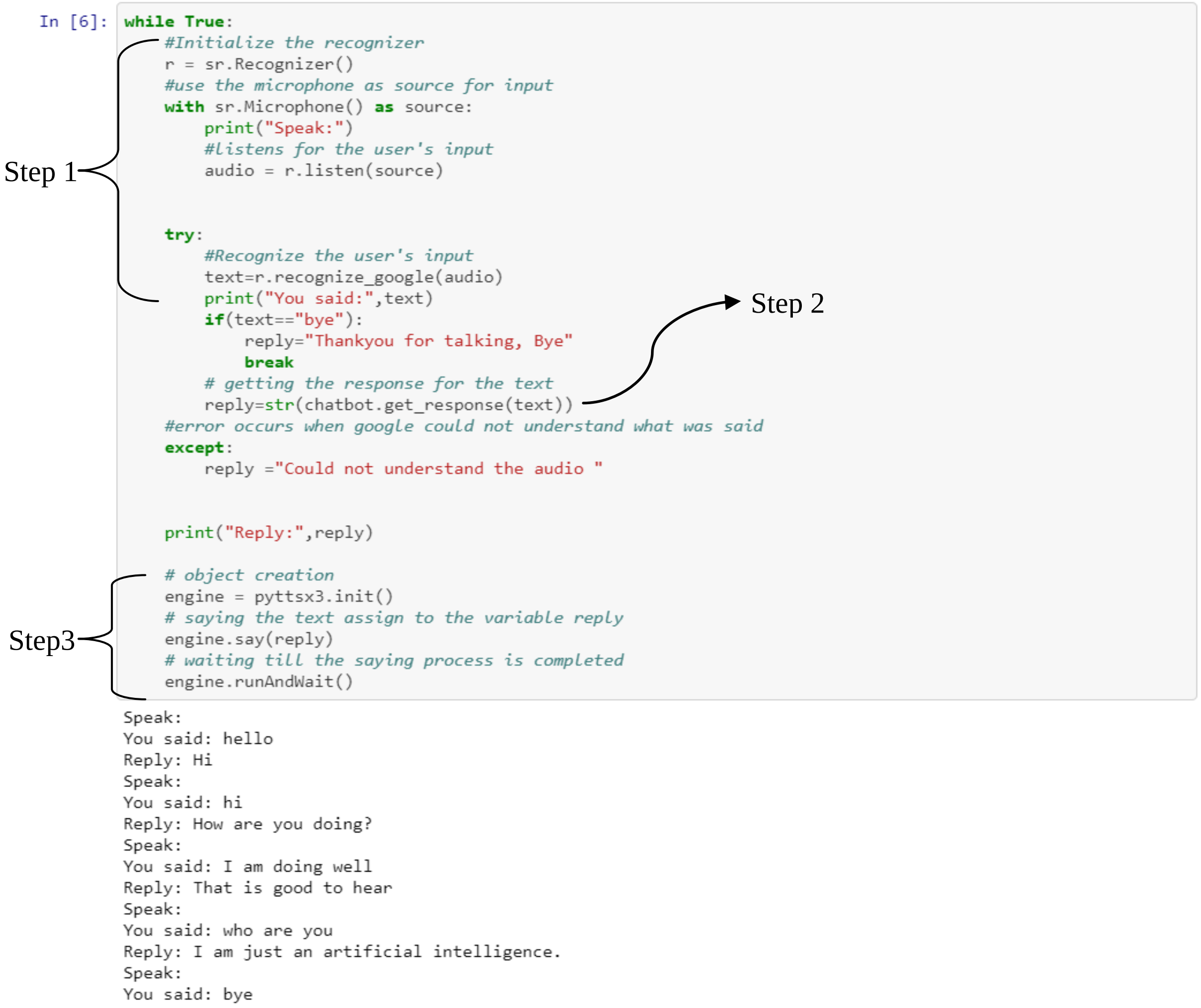


```
# Making an object of Chatbot and Setting the name of chatbot Mohan
chatbot = ChatBot('Mohan')

# Set the ChatterBotCorpusTrainer as trainer which allow to train the chatbot
chatbot.set_trainer(ChatterBotCorpusTrainer)

# Train the chatbot using data from the ChatterBot dialog corpus
chatbot.train("chatterbot.corpus.english")
```

```
ai.yml Training: [####################] 100%
botprofile.yml Training: [####################] 100%
computers.yml Training: [####################] 100%
conversations.yml Training: [####################] 100%
emotion.yml Training: [####################] 100%
food.yml Training: [####################] 100%
gossip.yml Training: [####################] 100%
greetings.yml Training: [####################] 100%
history.yml Training: [####################] 100%
humor.yml Training: [####################] 100%
literature.yml Training: [####################] 100%
money.yml Training: [####################] 100%
movies.yml Training: [####################] 100%
politics.yml Training: [####################] 100%
psychology.yml Training: [####################] 100%
science.yml Training: [####################] 100%
sports.yml Training: [####################] 100%
trivia.yml Training: [####################] 100%
```

Figure 7.3.2: Chatbot training with chatterbot corpus dataset

Voice Chatbot



```
In [6]: while True:
            #Initialize the recognizer
            r = sr.Recognizer()
            #use the microphone as source for input
            with sr.Microphone() as source:
                print("Speak:")
                #listens for the user's input
                audio = r.listen(source)


            try:
                #Recognize the user's input
                text=r.recognize_google(audio)
                print("You said:",text)
                if(text=="bye"):
                    reply="Thankyou for talking, Bye"
                    break
                # getting the response for the text
                reply=str(chatbot.get_response(text))
            #error occurs when google could not understand what was said
            except:
                reply ="Could not understand the audio "


            print("Reply:",reply)

            # object creation
            engine = pyttsx3.init()
            # saying the text assign to the variable reply
            engine.say(reply)
            # waiting till the saying process is completed
            engine.runAndWait()
```

```
Speak:
You said: hello
Reply: Hi
Speak:
You said: hi
Reply: How are you doing?
Speak:
You said: I am doing well
Reply: That is good to hear
Speak:
You said: who are you
Reply: I am just an artificial intelligence.
Speak:
You said: bye
```

Figure 7.3.3: Voice Chatbot which take voice as input and gives voice as output

In the Figure 7.3.1, we can see all the necessary packages required to build the voice chatbot.

In the Figure 7.3.2, we can see the training of the chatbot which we have to build before going through the steps.

In the Figure 7.3.3, we can see all the three steps inside the code and also we can see the input output of the voice chatbot. "You said" is the word or line spoken by the user as input and "Reply" is the word or line provided by the chatbot as output.

CHAPTER 8
Conclusion

During the Project Period from 6th January, 2020 to 26th June, 2020, I learned a lot in the field of Machine Leaning and Deep Learning. I learned something which I never learned before.

In this project, I have developed Convolutional Neural Network (CNN) model for **Traffic Sign Recognition (TSR)** to recognize the traffic sign images using German Traffic Sign Recognition Benchmark (GTSRB) Dataset. The motive for doing this work is to understand the concept of frame and neural network which have been used for Predictive Maintenance.

After TSR, I have developed Convolutional Neural Network (CNN), Long Short Term Memory (LSTM), CNN - LSTM and Convolutional LSTM (ConvLSTM) models for **Predictive Maintenance** and selected the best model based on accuracy and size to predict that which engine will need maintenance after 30 days using Turbofan Engine Degradation Simulation Dataset.

I have also developed a simple **Voice Chatbot** which will take voice as input and give voice as output.

On visual inspection the results are found satisfactory.

CHAPTER 9
Bibliography

[1] https://gyrus.ai/about_us

[2] https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign

[3] https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/

[4] https://www.upgrad.com/blog/top-python-libraries-for-machine-learning/

[5] https://medium.com/@himadrisankarchatterjee/a-basic-introduction-to-convolutional-neural-network-8e39019b27c4

[6] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[7] https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148

[8] https://medium.com/@purnasaigudikandula/recurrent-neural-networks-and-lstm-explained-7f51c7f6bbb9

[9] https://medium.com/analytics-vidhya/cnn-lstm-architecture-and-image-captioning-2351fc18e8d7

[10] https://medium.com/@MohammedS/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b#:~:text=Different%20performance%20metrics%20are%20used%20to%20evaluate%20different%20Machine%20Learning%20Algorithms.&text=We%20can%20use%20classification%20performance,(Area%20under%20Curve)%20etc.

[11] https://www.geeksforgeeks.org/confusion-matrix-machine-learning/#:~:text=A%20confusion%20matrix%20is%20a,the%20performance%20of%20an%20algorithm

[12] https://problemsolvingwithpython.com/01-Orientation/01.03-Installing-Anaconda-on-Windows/

[13] https://mas-dse.github.io/startup/anaconda-windows-install/

[14]https://expertsystem.com/chatbot/#:~:text=Why%20chatbots%20are%20important,typical%20cost%20of%20customer%20service.