**A PROJECT REPORT ON**

**EMERGING METHODS FOR EARLY DETECTION OF**

**FOREST FIRE**

# **Domain:** ARITIFICIAL INTELLIGENCE

**Team ID: PNT2022TMID22350**

Team Members:

# Raghu Rajagopal k (TL) (113119UG03078) Sanjaykumar s (113119UG03089)

# Dinesh kumar c (113119UG03021) Vasanth S (113119UG03111)

TABLE OF CONTENTS

1. INTRODUCTION.
   1. Project Overview.
   2. Purpose.
2. LITERATURE SURVEY.
   1. Existing Problem.
   2. References.
   3. Problem Statement Definition

3.IDEATION AND PROPOSED SOLUTION.

* 1. Empathy Map Canvas.
  2. Ideation and Brainstorming.
  3. Proposed Solution.
  4. Problem Solution Fit.

1. REQUIREMENT ANALYSIS.
   1. Functional Requirements.
   2. Non-Functional Requirements.
2. PROJECT DESIGN
   1. Data Flow Diagrams.
   2. Solution and Technical Architecture.
   3. User Stories.
3. PROJECT PLANNING AND SCHEDULING.
   1. Sprint Planning and Estimation
   2. Sprint Delivery Schedule.
   3. Reports from JIRA.
4. CODING AND SOLUTIONING
   1. Feature 1
   2. Feature 2
   3. Database Schema
5. TESTING
   1. Test Cases
   2. User Acceptance Testing
6. RESULTS
   1. Performance Metrics
7. ADVANTAGES AND DISADVANTAGES
8. CONCLUSION
9. FUTURE SCOPE
10. APPENDIX
    1. Source Code
    2. Git

## 1. INTRODUCTION

### 1.1 PROJECT OVERVIEW

Forest fires are a major environmental issue, creating economic and ecological damage while endangering human lives. There are typically about 100,000 wildfires in the United States every year. Over 9 million acres of land have been destroyed due to treacherous wildfires. It is difficult to predict and detect Forest Fire in a sparsely populated forest area and it is more difficult if the prediction is done using ground-based methods like Camera or Video-Based approach. Satellites can be an important source of data prior to and also during the Fire due to its reliability and efficiency. The various real-time forest fire detection and prediction approaches, with the goal of informing the local fire authorities

### 1.2 PURPOSE

Forest fires have become a major threat around the world, causing many negative impacts on human habitats and forest ecosystems. Climatic changes and the greenhouse effect are some of the consequences of such destruction. A higher percentage of forest fires occur due to human activities .The goal of the project is to develop a forest fire detection system that can identify forest fires in their early phases.

**The main goal of the project**

1. We can find forest fire early to avoid vulnerability and upcoming disaster.
2. Early Warning system to alert the officers and people to save lot of lives.
3. It is real time detection of forest fire.
4. To get most value accuracy.

# 2. LITERATURE SURVEY

## 2.1 Existing Problem

Every year, there are an estimated340,000 premature deaths from respiratory and cardiovascular issues attributed to wildfire smoke. The increasing frequency and severity of wildfires pose a growing threat to biodiversity globally. Individuals, companies and public authorities bear great economic costs due to fires. In order to reduce all these, we need to detect the forest fire at an

early stage and prevent it

## 2.2 References

* Torquay Celik , Huseyin Ozkaramanl, and Hassan Demirel (2007). Fire and Smoke detection without Sensors: Image Processing based approach.15th European signal processing conference (eusipco 2007), Poznan, Poland, September 3-7.
* S. A. Christopher, M. Wang, T. A. Berendes, and R. M. Welch (1998). The 1985 biomass burning season in South America: Satellite remote sensing of fires, smoke, and regional radiative energy budgets, vol. 37, 661– 678
* Paulo Vinicius Koerich Borges (2010). A Probabilistic Approach for VisionBased Fire Detection in Videos, IEEE transactions on circuits and systems for video technology, vol. 20, no. 5.
* Jiawei Han, Micheline Kamber, Jian Pei (2012). Data Mining Concepts and Techniques, Third edition, 248-253, 350-351.
* Official webpage of the European Forest Fire Information System at: http://effis.jrc.ec.europa.eu/
* Jesús San-Miguel-Ayanz, Tracy Durrant, Roberto Boca, Giorgio

Libertà, Alfredo Branco, Daniele de Rigo, Davide Ferrari,

Pieralberto Maianti, Tomàs Artés Vivancos, Hugo Costa, Fabio

Lana, Peter Löffler, Daniel Nuijten, Anders Christofer Ahlgren,

Thaïs Leray; Forest Fires in Europe, Middle East and North Africa 2017. EUR 29318 EN, ISBN 978-92-79-92831-4, doi: 10.2760/663443

* Chen, Thou-Ho, et al. "The smoke detection for early fire- alarming system base on video processing." Intelligent

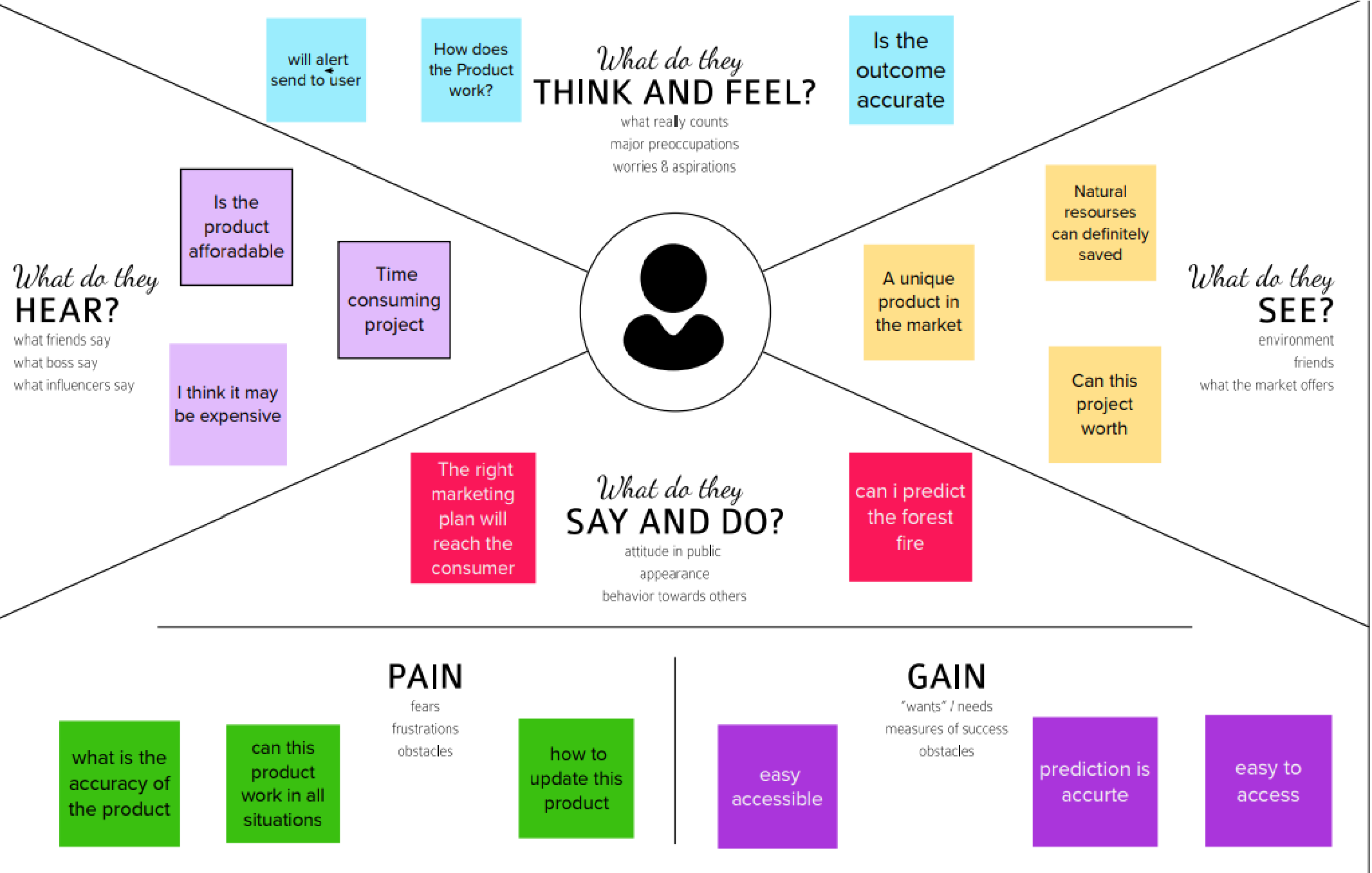
Information Hiding and Multimedia Signal Processing, 2006. IIH- MSP'06. International Conference on. IEEE, 2006.

Noda, S., and K. Ueda. "Fire detection in tunnels using an image processing method." Vehicle Navigation and Inform

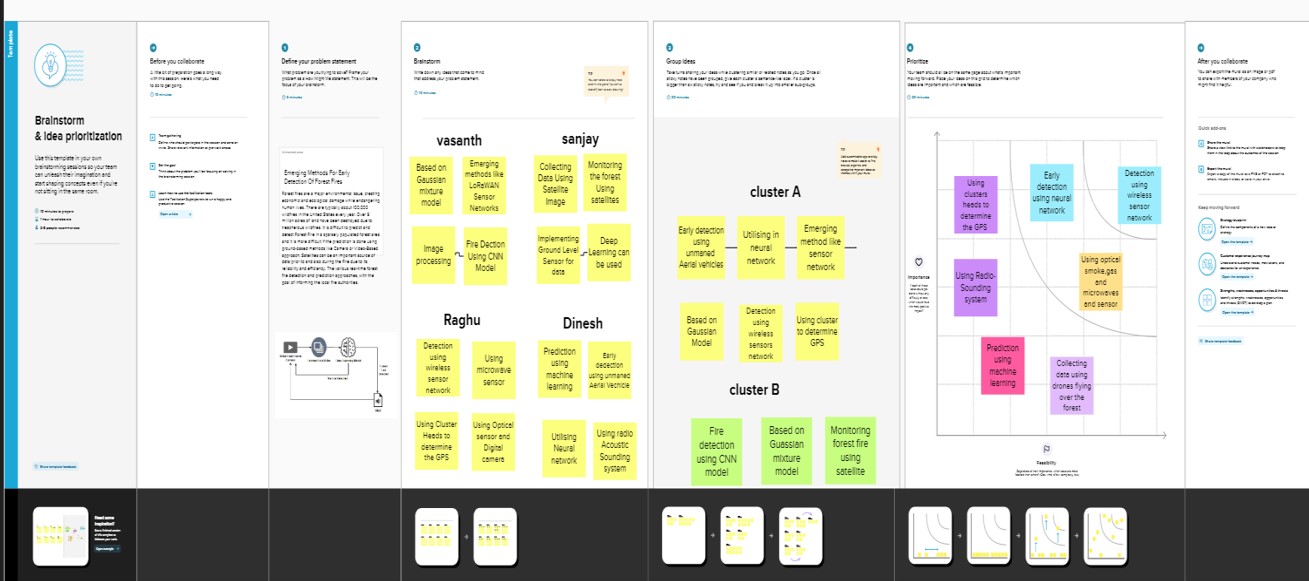
## 2.3 Problem Statement Definition

* In the past, fires were detected by watching towers or using satellite images.
* Satellites collect images of fires and send them to a monitoring authority for review. If the images 6 appear to show a fire, the authority will determine whether The fire is burning or not.
* But this approach was slow because the fire may have spread in the large areas and caused a lot of damage before the rescue team arrived.
* Since it's impossible to place a man in every part of a forest, it's important to have monitoring devices in certain areas so we can keep an eye on the forest.
* Both watching towers and satellite images failed to detect the presence of a fire early on, which resulted in more damage being done by the fire.
* Predictive analytics based on these insights are becoming increasingly effective in detecting mitigating and preventing fires.

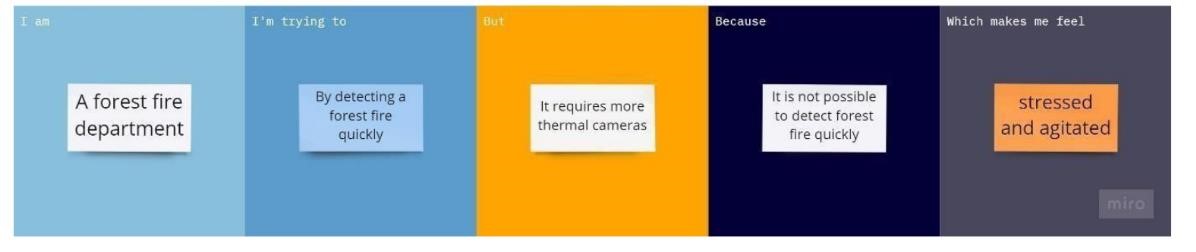
# 3. IDEATION AND PROPOSED SOLUTION 3.1 Empathy Map Canvas



## 3.2 Ideation and Brainstorming

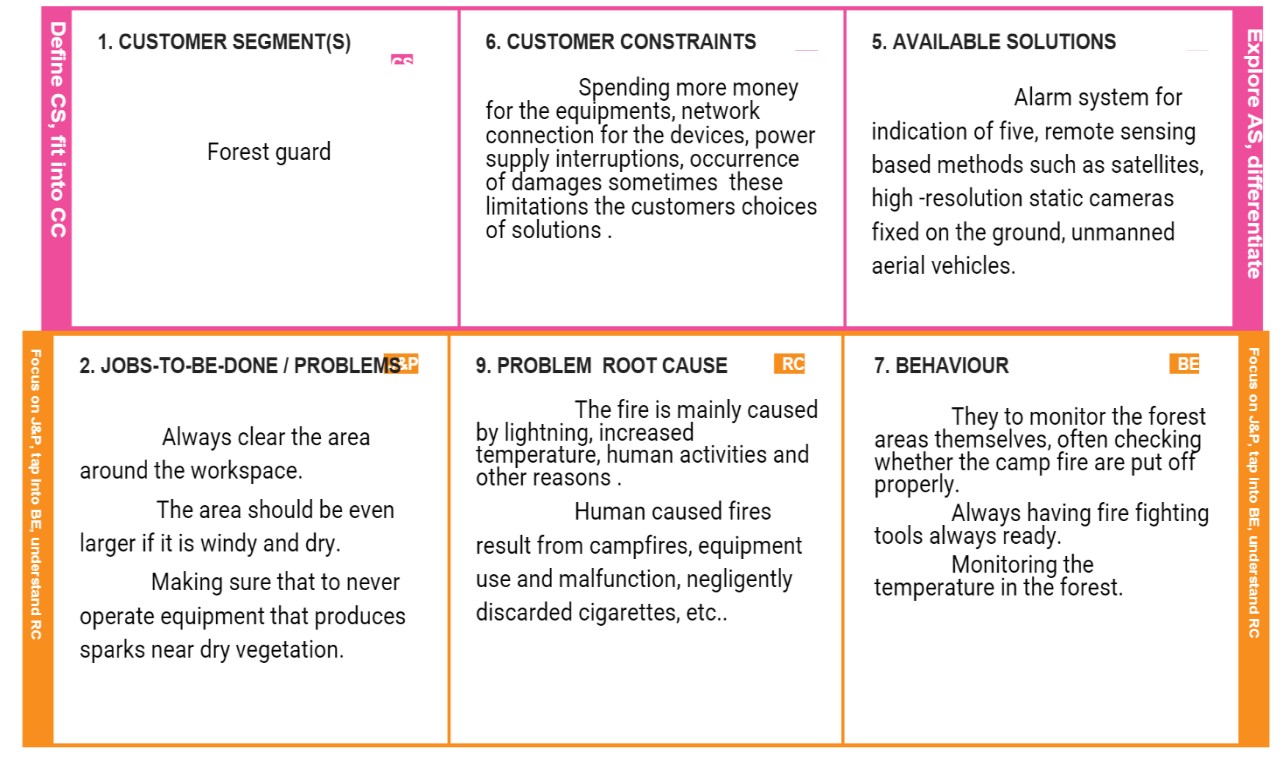


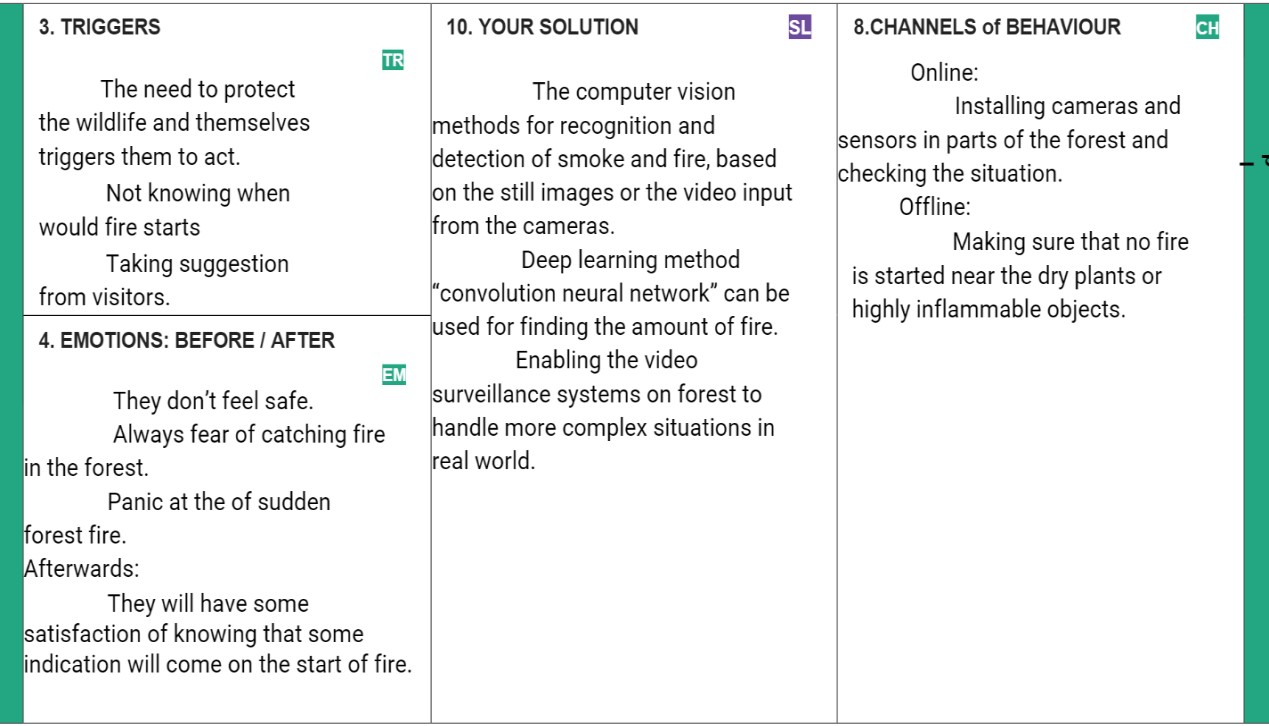
## 3.3 Proposed Solution



|  |  |  |
| --- | --- | --- |
| **S/no** | **Parameter** | Description |
| ➢ | Problem Statement (Problem to be solved) | A forest fire risk prediction algorithm, based on support vector machines, is presented. The algorithm depends on previous weather conditions in order to predict the fire hazard level of a day |
| ➢ | Idea / Solution description | Use computer vision methods for recognition and detection of smoke or fire, based on the still images or the video input from the drone camera |
| ➢ | Novelty / Uniqueness | Real time computer program detect forest fire in earliest before it spread to larger area. |
| ➢ | Impact on society | Blocked roads and railway lines, electricity, mobile and land telephone lines cut, destruction of homes and industry |
| ➢ | Business Model (Revenue Model) | The proposed method was implemented using the Python programming language on a Core i3 or greater ( CPU and 4GB RAM.) |
| ➢ | Scalability of the Solution | Changes in the use or occupancy of a building can result in compliance issues and a fire alarm system that no longer provides sufficient protection. If future changes are anticipated, fire safety engineers can design a fire alarm system with this in mind, providing a flexible infrastructure that includes the proper wire size and additional circuits distributed in a way that accommodates future growth and change can trigger potentially very expensive alterations in a fire alarm system. |

3.4 Problem Solution Fit





# 4. REQUIREMENT ANALYSIS

## 4.1 Functional Requirements

Following are the functional requirements of the proposed solution

|  |  |  |
| --- | --- | --- |
| FR NO | FUCTION  REQUIREMENT(EPIC  ) | SUB REQUIRMENT |
| ❖ | Video surveillance start | Start surveillance through remote control |
| ❖ | Forest monitoring | Continuous monitoring through camera |
| ❖ | Detect fire | Fire is detected through CNN ,odel |
| ❖ | Alert | Alert the forest officials through message |

### 4.2 NON- FUNCTIONAL REQUIREMENTS

Following are the non-functional requirement of the proposed solution.

|  |  |  |
| --- | --- | --- |
| FR no | FUCTION REQUIREMENT(EPIC) | SUB REQUIRMENT |
| ❖ | Reliability | Model is safe to install |
| ❖ | Security | more secure environment |
| ❖ | Availability | Build model is available all the time |
| ❖ | performance | Model will achieve high accuracy |

4.3 SYSTEM REQUIREMENTS:

The hardware requirements may serve as the basis for contract for the implementation of the system and should therefore be a complete engineer a the starting point for thE system design. Ram : 8GB Ram or more

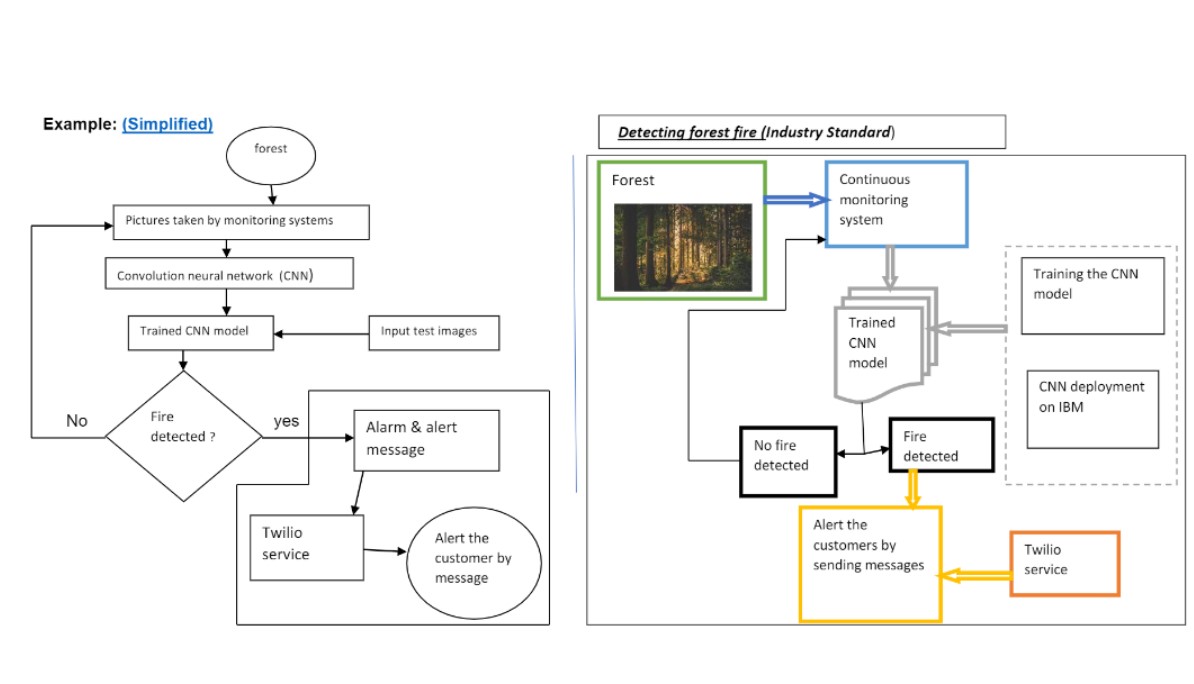
Processor : Any Processor

GPU : 8GB or more

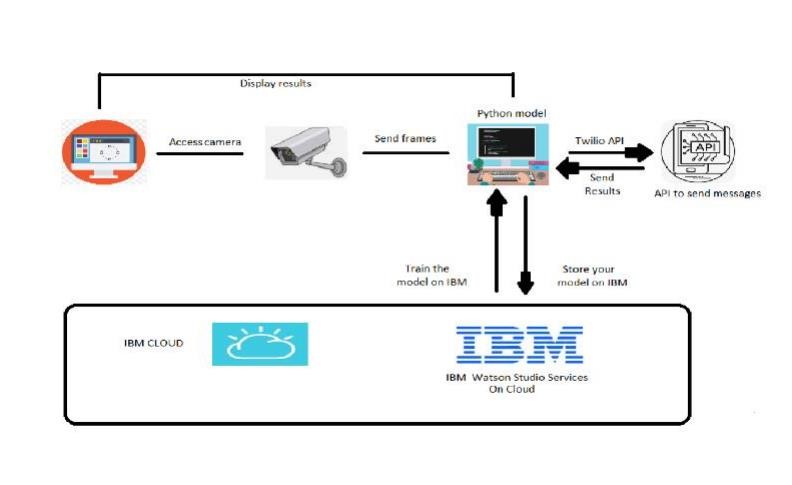
Hard Disk : 10GB or more

Speed : 1.4GHZ or more

# 5. PROJECT DESIGN 5.1Data Flow Diagrams

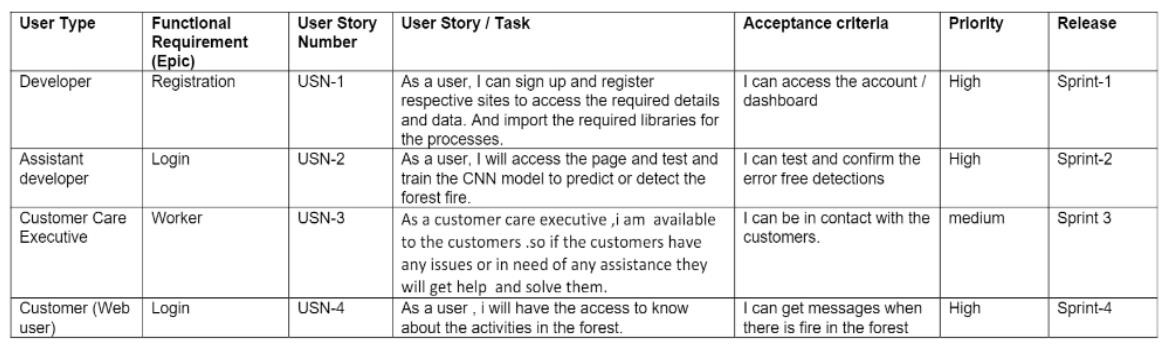


5.2 Solution & Technical Architecture:



:

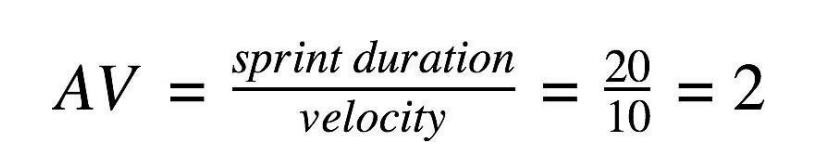
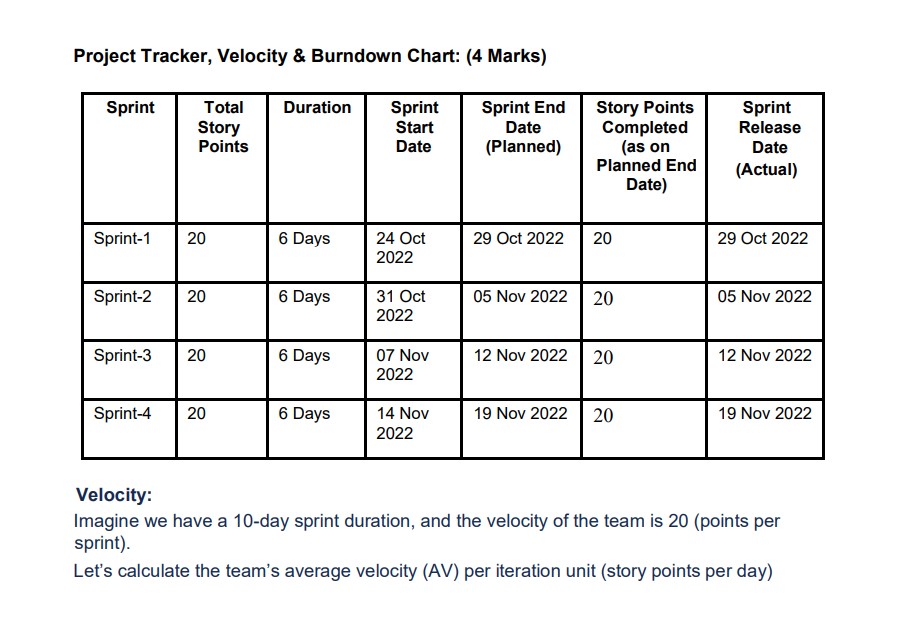
5.3 User Stories:



# 6.PROJECT PLANNING & SCHEDULING

## 6.1 Sprint Planning & Estimation

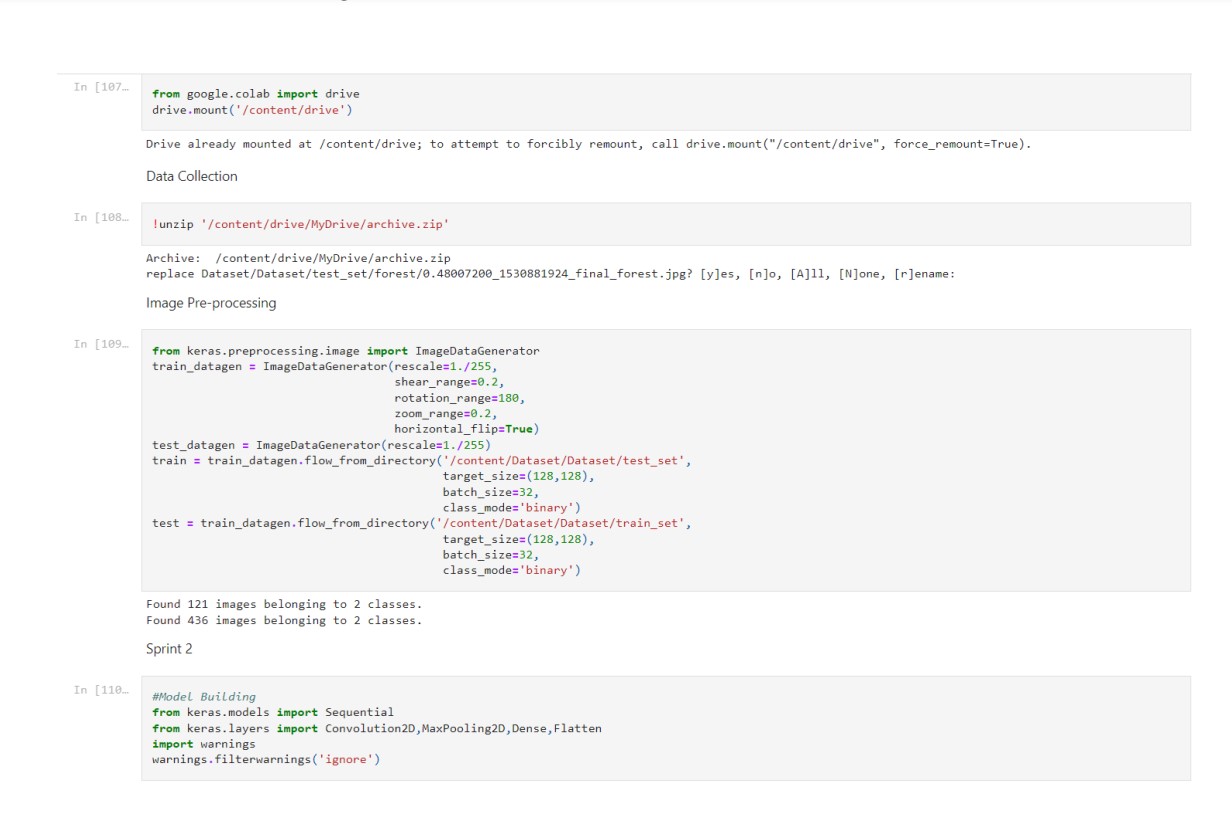
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint | Functional  Requirement (Epic) | User  Story  Number | User Story / Task | Story Points | Priorit  y | Team Members |
| Sprint1 | registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | 10 | high | Dinesh kumar  c |
| Sprint2 |  | USN-2 | As a user, I can register for the application through g mail , linked in | 10 | high | Vasanth s |
| Sprint3 | login | USN-2 | As a user ,I can login by using valid user name and password. | 20 | high | Sanjaykumar s |
|  |  |  | As a user ,I can view the garbage storage level. | 20 | high | Raghu  rajagopal k |
|  |  |  | Blynk Server is responsible for all the  communications between the smart phone and hardware. | 20 | high | Raghu  rajagopal k |

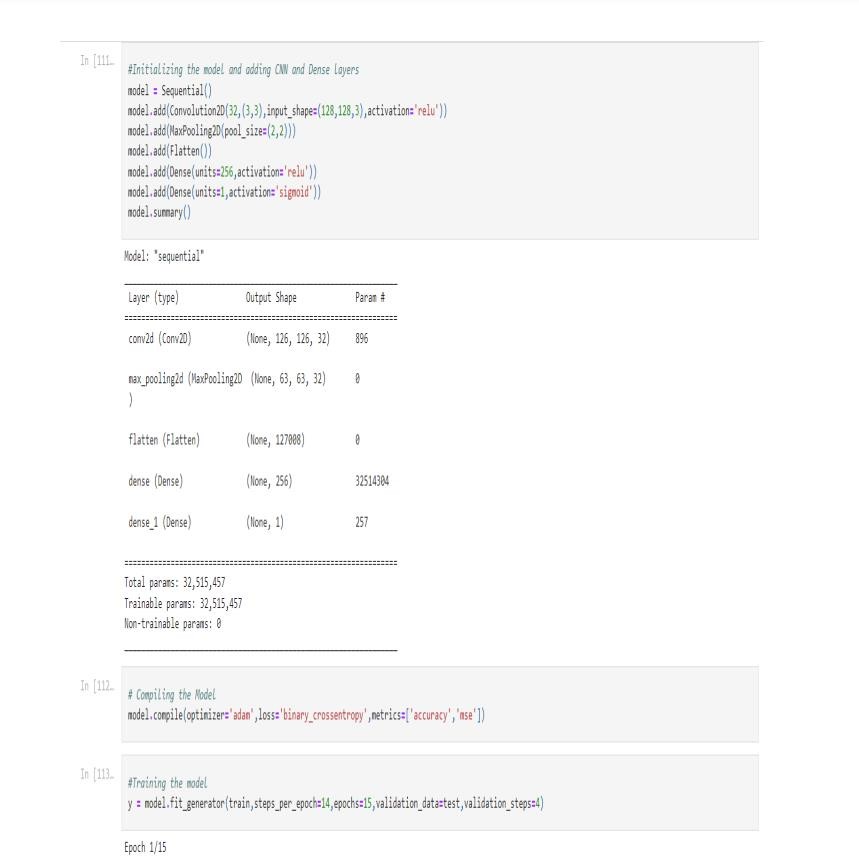


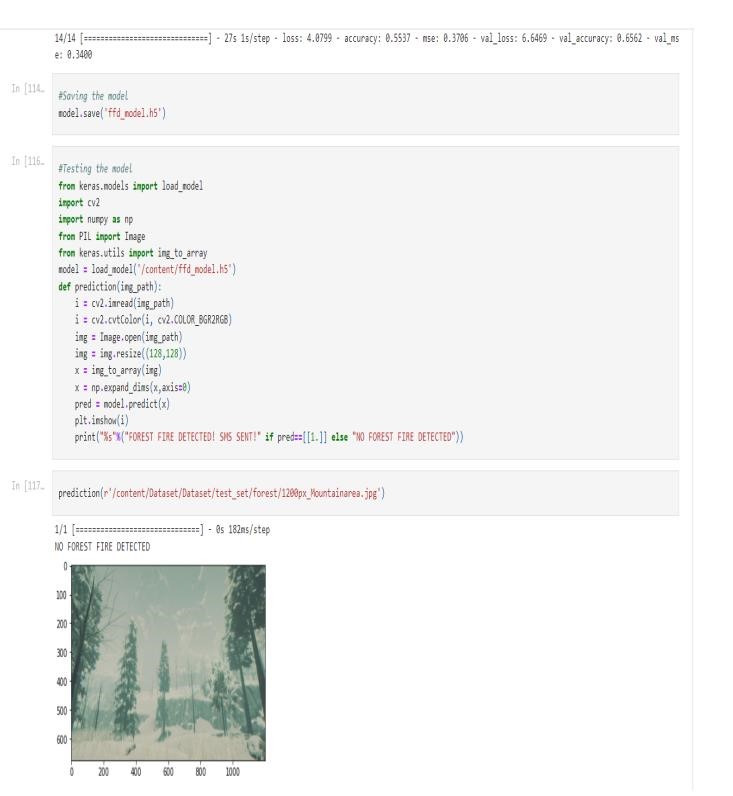
# 7. CODING AND SOLUTION

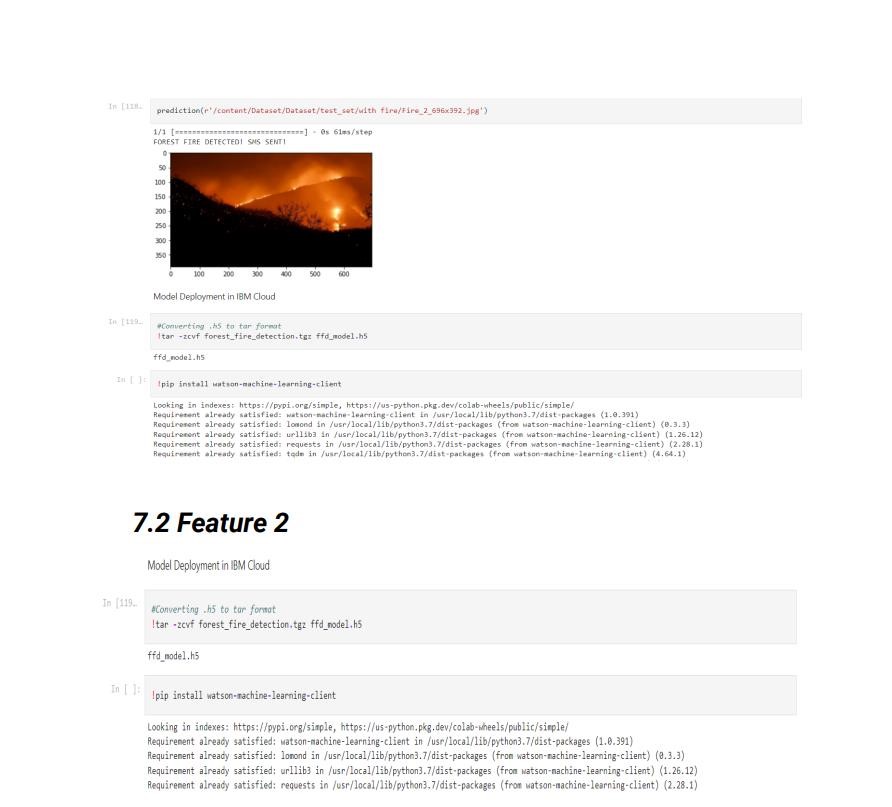
## 7.1 Feature 1

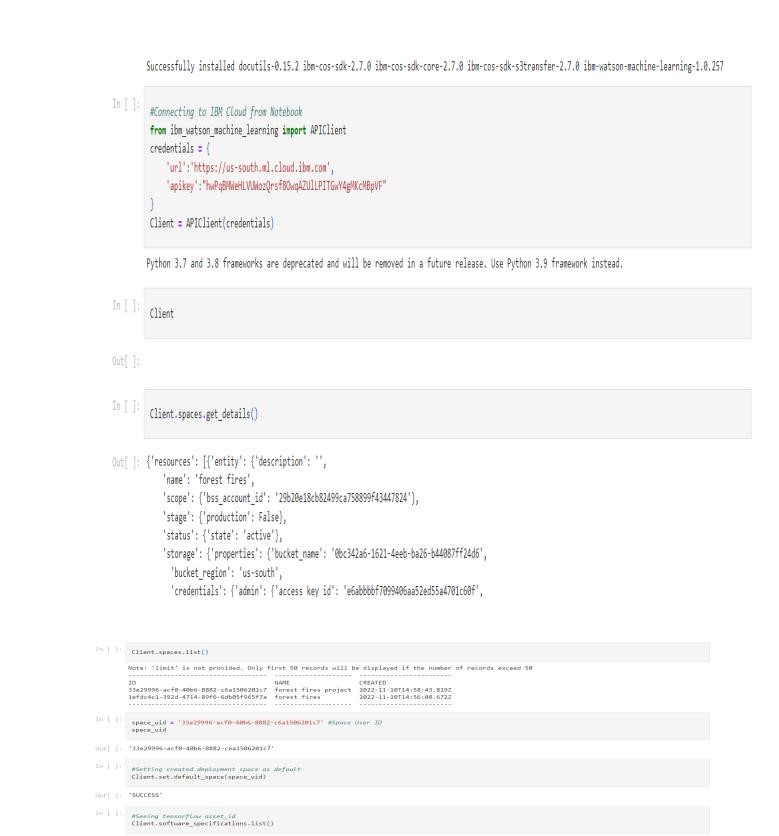
* Language used: Python
* Tools/IDE: Google Co lab

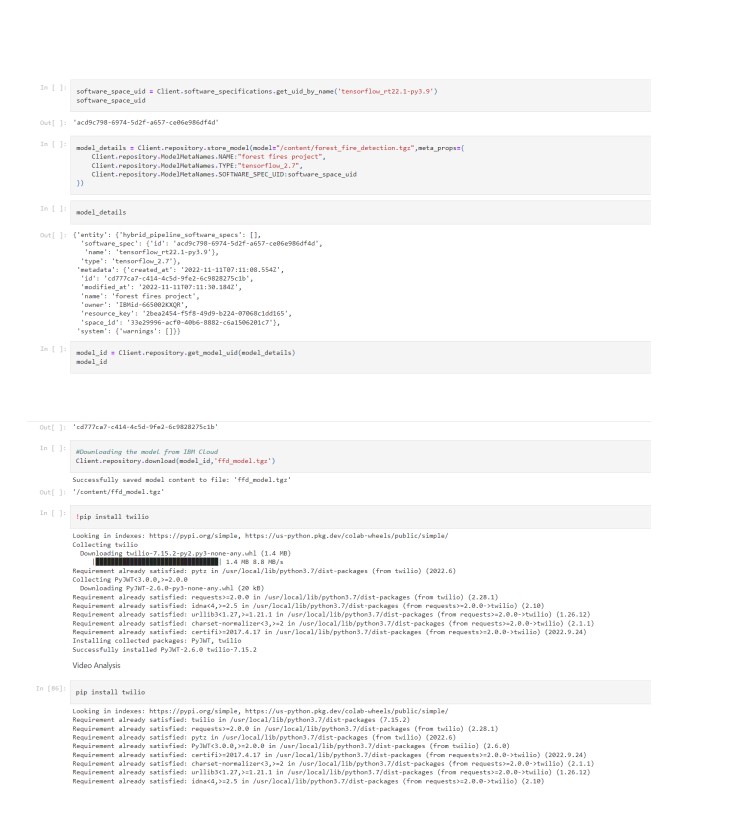












# 8. TESTING

## 8.1 Test Cases

A test case might be created as an automated script to verify the functionality per the original acceptance criteria. After doing manual exploratory testing, QA testers might suggest other functionality be added to the application as well as updated test cases be incorporated in the automated test suit

|  |  |  |  |
| --- | --- | --- | --- |
| TEST CASE ED | FEATURE TYPE | COMPONENT | TEST SCENARIO |
| SMS  NOTIFICATION | Twilio SMS  Notification | Python | Verify if user is able to receive SMS when forest fire is detected in the video processed by the model |
| DEPLOYMENT  TC | Website  Deloyment | Azure | Website built using HTML and designed using CSS is deployed using Microsoft Azure |
| FRONT END AND BACKEND TC | Website  Functionality | Home page  (client) | Verify if front-  end and backend are well conn ected and  the results are as expected |

Table.8.2. Test Report

|  |  |  |  |
| --- | --- | --- | --- |
| STEPS TO EXECUTE | TEST DATA | RESULT | EXECUTED BY |
| 1. Execute script with intended video file to check. 2. Check if the results are expected. 3. Note the results | User should receive SMS Notification | PASS | SANJAYKUMAR S |
| 1. Deploy repository on Azure 2. Check if the website is live. 3. Note the result | Website should be live in the given URL | PASS | SANJAYKUMAR S.  RAGHU  RAJAGOPAL.K |
| 1. Upload an image to the website to check the model’s working 2. Check if the results are as expected. 3. Note the results. | We should get expected detection  results for the uploaded image | PASS | RAGHU  RAJAGOPAL.K |

## 8.2 User Acceptance Testing Test Scenario

Predict the Output.

**Description**: To predict the output for the given input video or image.

**Test Step: Model**: 1. Choose Video file or use default video or use webcam input.

1. Execute the program.
2. If fire detected user receives SMS alert and console also displays and sounds an alert.

**Website:** 1. Choose Image file as input.

1. Click upload .
2. Website shows the result – ‘Positive’ or ‘Negative’ . Expected Result : Should display the exact prediction.

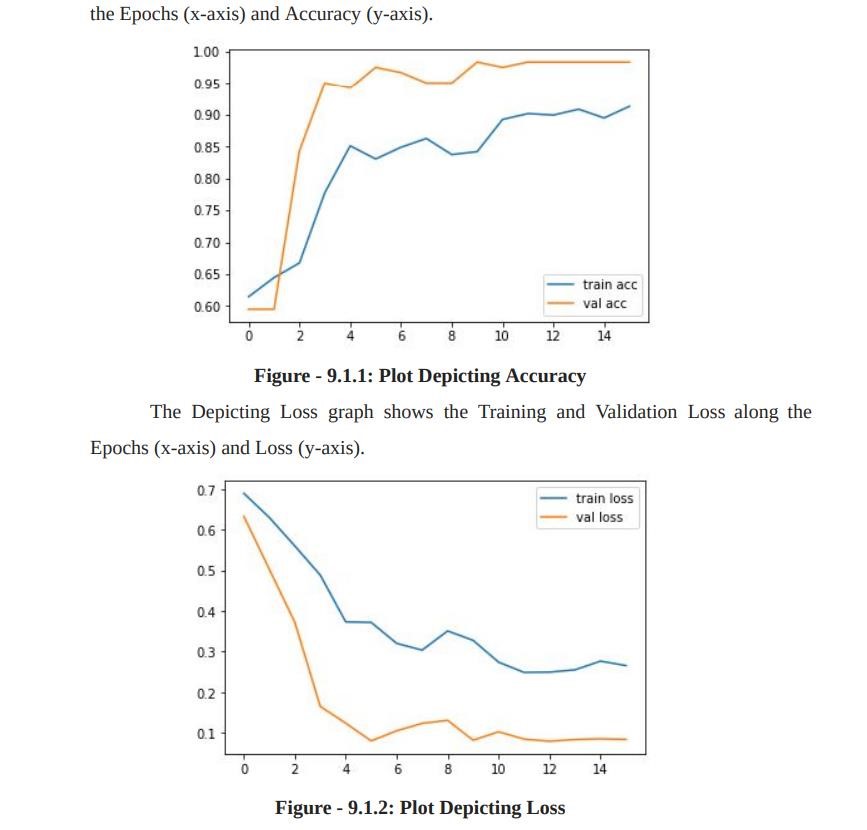
Actual Result: As Expected.

Status: PASS.

# 9.RESULTS

Performance Metrics The Depicting Accuracy graph shows the Training and Validation

Accuracy along the Epochs (x-axis) and Accuracy (y-axis)



## ADVANTAGES AND DISADVANTAGES ADVANTAGES

* This project helps forest officers and fire fighters to respond quickly to the forest fire so that they can handle it in its earlier phases.
* This project can be scaled to a large area easily given that there are a few basic requirements fulfilled. ● The detection accuracy obtained in this project is much better as compared to the existing methodologies used. It can produce significant results and with even more live data available to train, the model can be improved much more decreasing false results

. DISADVANTAGES

* The current version of the project cannot handle large amount of data for processing i.e., detecting the forest fire.
* There is no clear graphical user interface to handle and organize all video input efficiently.

# CONCLUSION

The project mainly helps forest officers and fire fighters to prevent forest fires and stop the forest fires from spreading. It also helps police officers and environmentalists to support in the rescue process. Forest fires pose a great threat to the environment, they decrease the quality of forests, endanger many species of flora and fauna resulting in depletion of natural resources and loss of human lives.

The current response time for handling the forest fires is too long. The delay in response can cause a fatal accident and it also increases the probability of the fire spreading wider. So, this project detects forest fire using Deep Learning and immediately alerts responsible people with SMS alert. It aims to decrease the response time to limit the damage by fighting the fire in its weak beginning phase.

FUTURE SCOPE

The current version of this project sends SMS alert to a single registered number using Twilio API. It also has a lower video processing capacity – to both capture and to detect forest fires. Some other additional features that are planned to be incorporated with this existing product are listed below:

* User can fetch multiple live cam input using a more powerful and robust processing system.
* User can use a latitude and longitude-based camera system to survey the forest area completely while scanning for animal movement to make sure of their presence in the region. ● User can also use UAV or drones in our response team to assist the fire fighters while also capturing real-time data.
* User can also create a more enhanced dashboard with more than binary response, we can include live temperature and natural gas levels (caused by decomposing material) and a quick response system to improve efficiency and decrease response time.

from google.colab import drive drive.mount('/gdrive')

Mounted at /gdrive

!wget https://github.com/DeepQuestAI/Fire-Smoke-Dataset/releases/download/v1/FIRE-SMOKE-DA

!unzip FIRE-SMOKE-DATASET.zip

--2022-11-18 15:04:09-- [https://github.com/DeepQuestAI/Fire-Smoke-Dataset/releas](https://github.com/DeepQuestAI/Fire-Smoke-Dataset/releases/download/v1/FIRE-SMOKE-DATASET.zip)

|  |
| --- |
|  |
|  |

Resolving github.com (github.com)... 20.27.177.113

Connecting to github.com (github.com)|20.27.177.113|:443... connected.

HTTP request sent, awaiting response... 302 Found

Location: [https://objects.githubusercontent.com/github-production-release-asset-2](https://objects.githubusercontent.com/github-production-release-asset-2e65be/193940929/09220a00-9842-11e9-8756-2d8df8631bb5?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20221118%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20221118T150409Z&X-Amz-Expires=300&X-Amz-Signature=cdfeac383979a9a6d1ffb01b2e723c3d5a5f07074c0fd62e723ba47305b44345&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=193940929&response-content-disposition=attachment%3B%20filename%3DFIRE-SMOKE-DATASET.zip&response-content-type=application%2Foctet-stream)

--2022-11-18 15:04:09-- [https://objects.githubusercontent.com/github-production](https://objects.githubusercontent.com/github-production-release-asset-2e65be/193940929/09220a00-9842-11e9-8756-2d8df8631bb5?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWNJYAX4CSVEH53A%2F20221118%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20221118T150409Z&X-Amz-Expires=300&X-Amz-Signature=cdfeac383979a9a6d1ffb01b2e723c3d5a5f07074c0fd62e723ba47305b44345&X-Amz-SignedHeaders=host&actor_id=0&key_id=0&repo_id=193940929&response-content-disposition=attachment%3B%20filename%3DFIRE-SMOKE-DATASET.zip&response-content-type=application%2Foctet-stream)Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.19

Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.1

HTTP request sent, awaiting response... 200 OK

Length: 320963592 (306M) [application/octet-stream]SavinSga..v. ing to: ‘FIRE-SMOKE-DATASET.zip’



FIRE-SMOKE-DATASET. 100%[===================>] 306.09M 5.95MB/s in 76s

2022-11-18 15:05:26 (4.00 MB/s) - ‘FIRE-SMOKE-DATASET.zip’ saved [320963592/32096

Archive: FIRE-SMOKE-DATASET.zipcreating: FIRE-SMOKE-DATASET/Test/creating: FIRE-SMOKE-DATASET/Test/Fire/inflating: FIRE-SMOKE-DATASET/Test/Fire/image\_0.jpg inflating: FIRE-SMOKE-DATASET/Test/Fire/image\_1.jpg inflating: FIRE-SMOKE-DATASET/Test/Fire/image\_10.jpg inflating: FIRE-SMOKE-DATASET/Test/Fire/image\_11.jpg inflating: FIRE-SMOKE-DATASET/Test/Fire/image\_12.jpg inflating: FIRE-SMOKE-DATASET/Test/Fire/image\_13.jpg inflating: FIRE-SMOKE-DATASET/Test/Fire/image\_14.jpg inflating: FIRE-SMOKE-DATASET/Test/Fire/image\_15.jpg inflating: FIRE-SMOKE-DATASET/Test/Fire/image\_16.jpg inflating: FIRE-SMOKE-DATASET/Test/Fire/image\_17.jpg inflating: FIRE-SMOKE-DATASET/Test/Fire/image\_18.jpg inflating: FIRE-SMOKE-DATASET/Test/Fire/image\_19.jpg inflating: FIRE-SMOKE-DATASET/Test/Fire/image\_2.jpg inflating: FIRE-SMOKE-DATASET/Test/Fire/image\_20.jpg inflating: FIRE-SMOKE-DATASET/Test/Fire/image\_21.jpg inflating: FIRE-SMOKE-DATASET/Test/Fire/image\_22.jpg

inflating: FIRE-SMOKE-DATASET/Test/Fire/image\_23.jpg

: FIRE-SMOKE-DATASET/Test/Fire/image\_24.jpg inflating: FIRE-SMOKE-

DATASET/Test/Fire/image\_25.jpg inflating: FIRE-SMOKE-

DATASET/Test/Fire/image\_26.jpg inflating: FIRE-SMOKE-

DATASET/Test/Fire/image\_27.jpg inflating: FIRE-SMOKE-

DATASET/Test/Fire/image\_28.jpg inflating: FIRE-SMOKE-

DATASET/Test/Fire/image\_29.jpg inflating: FIRE-SMOKE-

DATASET/Test/Fire/image\_3.jpg inflating: FIRE-SMOKEDATASET/Test/Fire/image\_30.jpg inflating: FIRE-SMOKE-

DATASET/Test/Fire/image\_31.jpg inflating: FIRE-SMOKE-

DATASET/Test/Fire/image\_32.jpg inflating: FIRE-SMOKE-

DATASET/Test/Fire/image\_33.jpg inflating: FIRE-SMOKE-

DATASET/Test/Fire/image\_34.jpg inflating: FIRE-SMOKE-

DATASET/Test/Fire/image\_35.jpg inflating: FIRE-SMOKE-

DATASET/Test/Fire/image\_36.jpg inflating: FIRE-SMOKE-

DATASET/Test/Fire/image\_37.jpg inflating: FIRE-SMOKE-

DATASET/Test/Fire/image\_38.jpg inflating: FIRE-SMOKE-

DATASET/Test/Fire/image\_39.jpg inflating: FIRE-SMOKE-

DATASET/Test/Fire/image\_4.jpg inflating: FIRE-SMOKEDATASET/Test/Fire/image\_40.jpg inflating: FIRE-SMOKEDATASET/Test/Fire/image\_41.jpg

import shutilshutil.rmtree('/content/FIRE-SMOKE-DATASET/Test/Smoke') shutil.rmtree('/content/FIRE-SMOKE-DATASET/Train/Smoke')

import tensorflow as tfimport keras\_preprocessingfrom keras\_preprocessing import imagefrom keras\_preprocessing.image import ImageDataGenerator import shutil

TRAINING\_DIR="/content/FIRE-SMOKE-DATASET/Train"

Saving...

training\_datagen = ImageDataGenerator(rescale=1./255,zoom\_range=0.15,horizontal\_flip=True, fill\_mode='nearest')VALIDATION\_DIR="/content/FIRE-SMOKE-DATASET/Test"



validation\_datagen = ImageDataGenerator(rescale =1./255)

train\_generator = training\_datagen.flow\_from\_directory( TRAINING\_DIR,target\_size=(224,224), shuffle = True,class\_mode='categorical', batch\_size = 128

)

validation\_generator = validation\_datagen.flow\_from\_directory( VALIDATION\_DIR,target\_size=(224,224),class\_mode='categorical', shuffle = True,batch\_size=14

)

Found 1800 images belonging to 2 classes. Found 200 images belonging to 2 classes.

from tensorflow.keras.applications.inception\_v3 import InceptionV3 from tensorflow.keras.preprocessing import imagefrom tensorflow.keras.models import Model

from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Input, Dropout

input\_tensor = Input(shape=(224,224,3))

base\_model = InceptionV3(input\_tensor=input\_tensor,weights='imagenet',include\_top=False)

#adding a global spatial average pooling layer x = base\_model.outputx = GlobalAveragePooling2D()(x)x = Dense(2048,activation='relu')(x) x = Dropout(0.2)(x)predictions = Dense(2, activation='softmax')(x)model = Model(inputs=base\_model.input, outputs=predictions)

for layer in base\_model.layers:

layer.trainable = False

model.compile(optimizer='rmsprop', loss='categorical\_crossentropy',metrics=["acc"])



for layer in model.layers[:249]: layer.trainable = False

for layer in model.layers[249:]:

layer.trainable = True

from tensorflow.keras.optimizers import SGD

model.compile(optimizer=SGD(lr=0.0001, momentum=0.9), loss='categorical\_crossentropy',metr

class myCallback(tf.keras.callbacks.Callback):

def on\_epoch\_end(self,epoch,logs={}):

if(logs.get('val\_loss')<=0.1099 and logs.get('loss')<=0.1099):

print('\n\n Reached the Destination!')self.model.stop\_training = True

callbacks = myCallback()

history =model.fit( train\_generator,steps\_per\_epoch = 3,epochs= 30,validation\_data = validation\_generator, validation\_steps = 1,callbacks=[callbacks]

)

print(len(base\_model.layers))

/usr/local/lib/python3.7/dist-packages/keras/optimizers/optimizer\_v2/gradient\_descen super(SGD, self).init(name, \*\*kwargs)

Epoch 1/30

3/3 [==============================]- 72s22s/step- loss:0.2371- acc:0.9089- vEpoch 2/30

3/3 [==============================]- 65s21s/step- loss:0.1964- acc:0.9193- vEpoch 3/30

3/3 [==============================]- 62s20s/step- loss:0.1813- acc:0.9141- vEpoch 4/30

3/3 [==============================]- 68s23s/step- loss:0.2045- acc:0.9089- vEpoch 5/30

3/3 [==============================]- 65s22s/step- loss:0.1895- acc:0.9036- vEpoch 6/30

3/3 [==============================]- 65s22s/step- loss:0.1743- acc:0.9271- v

Epoch 7/30

3/3 [==============================]- 65s21s/step- loss:0.1461- acc:0.9427- vEpoch 8/30

3/3 [==============================]- 46s13s/step- loss:0.1794- acc:0.9318- vEpoch 9/30

3/3 [==============================]- 65s21s/step- loss:0.1270- acc:0.9531- vEpoch 10/30

3/3 [==============================]- 67s21s/step- loss:0.1712- acc:0.9479- v

Epoch 11/30

3/3 [==============================]- 63s21s/step- loss:0.1357- acc:0.9505- v

Epoch 12/30

3/3 [==============================]- 65s22s/step- loss:0.1483- acc:0.9505- vEpoch 13/30

3/3 [==============================]- 67s23s/step- loss:0.1355- acc:0.9375- vEpoch 14/30

3/3 [==============================]- 47s22s/step- loss:0.0927- acc:0.9659- vEpoch 15/30

3/3 [==============================] - ETA: 0s - loss: 0.1027 - acc: 0.9609

Reached the Destination!

3/3 [==============================] - 65s 22s/step - loss: 0.1027 - acc: 0.9609 - v

311

model.save("model.h5")

NameError Traceback (most recent call last)

<ipython-input-9-a3439455f9ca>in <module>

----> 1 model.save("model.h5")

NameError: name 'model' is not defined



SEARCH

STACK

OVERFLOW

%matplotlib inlineimport matplotlib.pyplot as plt acc = history.history['acc']val\_acc= history.history['val\_acc'] loss = history.history['loss']val\_loss = history.history['val\_loss'] epochs = range(len(acc))

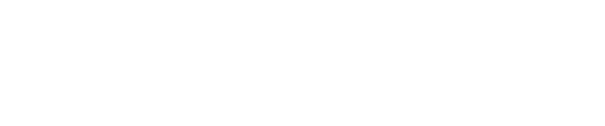
plt.plot(epochs,acc,'g',label='Training accuracy')plt.plot(epochs, val\_acc,'b', label='validation accuracy') plt.title('Training and validation accuracy')

plt.legend(loc=0) plt.figure()

plt.show()

plt.plot(epochs, loss, 'r', label = 'Training loss')plt.plot(epochs,val\_loss,'orange',label='Validation loss') plt.title('Training and Validation loss')

plt.legend(loc=0) plt.figure()plt.show()



Saving...

D ownloading PyJWT-2.6.0-py3-none-any.whl (20 kB)

<

Figure

size

432x288

with

0

Axes>

Figure

<

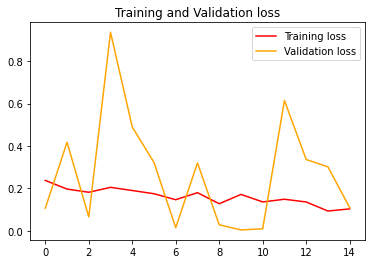
size

432x288

with

0

Axes>



from google.colab import drive

drive.mount('/content/drive')

newModel

=

tf.keras.models.load\_model("model.h5")

!

pip

install twilio

Looking in indexes:

[e](https://pypi.org/simple)

[https://pypi.org/simpl](https://pypi.org/simple)

[,](https://pypi.org/simple)

[s](https://us-python.pkg.dev/colab-wheels/public/simple/)

[https://u](https://us-python.pkg.dev/colab-wheels/public/simple/)

[-](https://us-python.pkg.dev/colab-wheels/public/simple/)

[python.pkg.dev/cola](https://us-python.pkg.dev/colab-wheels/public/simple/)

[b](https://us-python.pkg.dev/colab-wheels/public/simple/)

[-](https://us-python.pkg.dev/colab-wheels/public/simple/)

[wheels](https://us-python.pkg.dev/colab-wheels/public/simple/)

[/](https://us-python.pkg.dev/colab-wheels/public/simple/)

Collecting twilio

Downloading twilio

-

7.15.3

-

py2.py3

-

none

-

any.whl (1.4 MB)

|██

██████████████████████████████| 1.4 MB 4.7 MB/s

Requirement already satisfied: requests>=2.0.0 in /usr/local/lib/python3.7/dist

-

pack

Collecting PyJWT<3.0.0,>=2.0.0

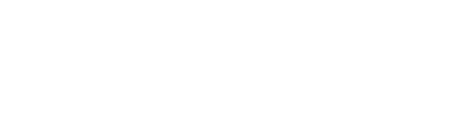
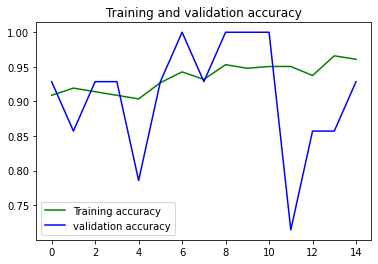
Saving..

.

Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist

-

pa



from

google.colab

import

drive

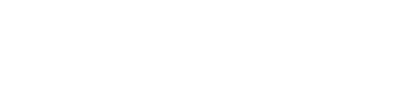
drive.mount('/gdrive')



import

tensorflow

as tf



%cd

/content/drive/MyDrive

Requirement already satisfied: pytz in /usr/local/lib/python3.7/dist-packages (from

Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-package

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-p

Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local

Installing collected packages: PyJWT, twilio

Successfully installed PyJWT-2.6.0 twilio-7.15.3

import os

from twilio.rest import Client

from traitlets.traitlets import ClassTypes import keras.utils as image#predicting any random images

import numpy as npfrom google.colab import files

tf.keras.utils.load\_img

#from keras.preprocessing import image

uploaded = files.upload() for fn in uploaded.keys(): # path= '/content/'+fnpath = fnimg = image.load\_img(path, target\_size=(224,224)) x = image.img\_to\_array(img)x=np.expand\_dims(x, axis=0)/255 classes = newModel.predict(x) if np.argmax(classes[0])==0:

print(np.argmax(classes[0])==0, max(classes[0]),end=" ") print("Forest Fire is detected !!!,Message sended")

# Sending Message to authority

account\_sid = 'ACb65b5505be868c24bd20543207a856a1' auth\_token = '0e306096c14f4fb7cce1d6536c09b3b2'client = Client(account\_sid, auth\_token)

message = client.messages.create(body='Forest Fire Detected !! Be Aware, precaustion needed move to safe place----from\_='+19182624326', to='+919361092334'

)

else:

print(np.argmax(classes[0])==0, max(classes[0]),end=" ") print("No forest fire is detected!!!")

Choose Filesth (2).jfif**th (2).jfif**(image/jpeg) - 25071 bytes, last modified: 8/15/2022 - 100% done



SavinSga..v. ing th (2).jfif to th (2) (10).jfif

1/1 [==============================] - 0s 345ms/step

True 0.9402907 Forest Fire is detected !!!,Message sended

!pip install watson-machine-learning-client

Looking in indexes: [https://pypi.org/simple,](https://pypi.org/simple)[https://us-python.pkg.dev/colab-whee](https://us-python.pkg.dev/colab-wheels/public/simple/)



Collecting watson-machine-learning-client

Downloading watson\_machine\_learning\_client-1.0.391-py3-none-any.whl (538 kB)

|████████████████████████████████| 538 kB 4.8 MB/s

Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages

Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (

Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (fr Collecting ibm-cos-sdk

Downloading ibm-cos-sdk-2.12.0.tar.gz (55 kB)

|████████████████████████████████| 55 kB 3.7 MB/s Collecting lomond

Downloading lomond-0.3.3-py2.py3-none-any.whl (35 kB)

Collecting boto3

Downloading boto3-1.26.13-py3-none-any.whl (132 kB)

|████████████████████████████████| 132 kB 56.0 MB/s

Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages

Requirement already satisfied: tabulate in /usr/local/lib/python3.7/dist-packages Collecting jmespath<2.0.0,>=0.7.1

|  |
| --- |
|  |
|  |

Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)

Collecting botocore<1.30.0,>=1.29.13

Downloading botocore-1.29.13-py3-none-any.whl (9.9 MB)

|████████████████████████████████| 9.9 MB 61.1 MB/s Collecting s3transfer<0.7.0,>=0.6.0

Downloading s3transfer-0.6.0-py3-none-any.whl (79 kB)

|████████████████████████████████| 79 kB 8.1 MB/s

Collecting urllib3

Downloading urllib3-1.26.12-py2.py3-none-any.whl (140 kB)

|████████████████████████████████| 140 kB 67.5 MB/s

Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/pyth

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages

Collecting ibm-cos-sdk-core==2.12.0

Downloading ibm-cos-sdk-core-2.12.0.tar.gz (956 kB)

|████████████████████████████████| 956 kB 56.7 MB/s Collecting ibm-cos-sdk-s3transfer==2.12.0

Downloading ibm-cos-sdk-s3transfer-2.12.0.tar.gz (135 kB)

|████████████████████████████████| 135 kB 47.9 MB/s Collecting jmespath<2.0.0,>=0.7.1

Downloading jmespath-0.10.0-py2.py3-none-any.whl (24 kB)

Collecting requests

Downloading requests-2.28.1-py3-none-any.whl (62 kB)

|████████████████████████████████| 62 kB 1.4 MB/s

Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-pack Requirement already satisfied: charset-normalizer<3,>=2 in /usr/local/lib/python3 Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-pac Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-pack

Building wheels for collected packages: ibm-cos-sdk, ibm-cos-sdk-core, ibm-cos-sdBuilding wheel for ibm-cos-sdk (setup.py) ... done

Created wheel for ibm-cos-sdk: filename=ibm\_cos\_sdk-2.12.0-py3-none-any.whl siz

Stored in directory: /root/.cache/pip/wheels/ec/94/29/2b57327cf00664b6614304f79 Building wheel for ibm-cos-sdk-core (setup.py) ... done

Created wheel for ibm-cos-sdk-core: filename=ibm\_cos\_sdk\_core-2.12.0-py3-none-a

Stored in directory: /root/.cache/pip/wheels/64/56/fb/5cd6f4f40406c828a5289b95b

Saving..B. uilding wheel for ibm-cos-sdk-s3transfer (setup.py) ... done

Created wheel for ibm-cos-sdk-s3transfer: filename=ibm\_cos\_sdk\_s3transfer-2.12.

Stored in directory: /root/.cache/pip/wheels/57/79/6a/ffe3370ed7ebc00604f9f7676 Successfully built ibm-cos-sdk ibm-cos-sdk-core ibm-cos-sdk-s3transfer



!pip install ibm\_watson-machine-learning

Looking in indexes: [https://pypi.org/simple,](https://pypi.org/simple)[https://us-python.pkg.dev/colab-whee](https://us-python.pkg.dev/colab-wheels/public/simple/)



Collecting ibm\_watson-machine-learning

Downloading ibm\_watson\_machine\_learning-1.0.257-py3-none-any.whl (1.8 MB)

|████████████████████████████████| 1.8 MB 4.9 MB/s

Requirement already satisfied: tabulate in /usr/local/lib/python3.7/dist-packages

Requirement already satisfied: lomond in /usr/local/lib/python3.7/dist-packages (

Requirement already satisfied: urllib3 in /usr/local/lib/python3.7/dist-packages

Collecting ibm-cos-sdk==2.7.\*

Downloading ibm-cos-sdk-2.7.0.tar.gz (51 kB)

|████████████████████████████████| 51 kB 674 kB/s

Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-package

Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages

Requirement already satisfied: certifi in /usr/local/lib/python3.7/dist-packages

Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/dis

Requirement already satisfied: pandas<1.5.0,>=0.24.2 in /usr/local/lib/python3.7/

Collecting ibm-cos-sdk-core==2.7.0

Downloading ibm-cos-sdk-core-2.7.0.tar.gz (824 kB)

|████████████████████████████████| 824 kB 53.1 MB/s

|  |
| --- |
|  |
|  |

Collecting ibm-cos-sdk-s3transfer==2.7.0



Downloading ibm-cos-sdk-s3transfer-2.7.0.tar.gz (133 kB)

|████████████████████████████████| 133 kB 58.3 MB/s

Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /usr/local/lib/python3.7

Collecting docutils<0.16,>=0.10

Downloading docutils-0.15.2-py3-none-any.whl (547 kB)

|████████████████████████████████| 547 kB 57.4 MB/s

Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/local/lib/pyth Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-pack Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-pac Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.7/dist-pack Requirement already satisfied: charset-normalizer<3,>=2 in /usr/local/lib/python3 Requirement already satisfied: typing-extensions>=3.6.4 in /usr/local/lib/python3 Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-package

Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3

Building wheels for collected packages: ibm-cos-sdk, ibm-cos-sdk-core, ibm-cos-sdBuilding wheel for ibm-cos-sdk (setup.py) ... done

Created wheel for ibm-cos-sdk: filename=ibm\_cos\_sdk-2.7.0-py2.py3-none-any.whl

Stored in directory: /root/.cache/pip/wheels/47/22/bf/e1154ff0f5de93cc477acd0ca

Building wheel for ibm-cos-sdk-core (setup.py) ... done

Created wheel for ibm-cos-sdk-core: filename=ibm\_cos\_sdk\_core-2.7.0-py2.py3-non

Stored in directory: /root/.cache/pip/wheels/6c/a2/e4/c16d02f809a3ea998e17cfd02 Building wheel for ibm-cos-sdk-s3transfer (setup.py) ... done

Created wheel for ibm-cos-sdk-s3transfer: filename=ibm\_cos\_sdk\_s3transfer-2.7.0

Stored in directory: /root/.cache/pip/wheels/5f/b7/14/fbe02bc1ef1af890650c7e517Successfully built ibm-cos-sdk ibm-cos-sdk-core ibm-cos-sdk-s3transfer

Installing collected packages: docutils, ibm-cos-sdk-core, ibm-cos-sdk-s3transfer

Attempting uninstall: docutils

Found existing installation: docutils 0.17.1

Uninstalling docutils-0.17.1:

Successfully uninstalled docutils-0.17.1

Attempting uninstall: ibm-cos-sdk-core

Found existing installation: ibm-cos-sdk-core 2.12.0Saving...Uninstalling ibm-cos-sdk-core-2.12.0:



Successfully uninstalled ibm-cos-sdk-core-2.12.0

Attempting uninstall: ibm-cos-sdk-s3transferFound existing installation: ibm-cos-sdks3transfer 2.12.0 Uninstalling ibm-cos-sdks3transfer-2.12.0:

#connecting to IBM cloudfrom ibm\_watson\_machine\_learning import APIClient credentials = {

"url":'https://us-south.ml.cloud.ibm.com',

"apikey": "xdlFwJU-rH3GFLdU6SvxYsJgRqsXlWur2-Y840Qes57R"

}

Client = APIClient(credentials)

Python 3.7 and 3.8 frameworks are deprecated and will be removed in a future release

Client

<ibm\_watson\_machine\_learning.client.APIClient at 0x7fe327fdcd10>

Github link :https://github.com/IBM-EPBL/IBM-Project-21862-1659793415

Demo link : https://youtu.be/rgukvo9p2Gk