EMERGING METHODS FOR EARLY DETECTION OF FOREST FIRES

NALAIYA THIRAN PROJECT BASED LEARNING

On

PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

A PROJECT REPORT

T.SRAVAN KUMAR 720819104115

A.JYOTHI SWAROOP 720819104005

B.NANDA KISHORE 720819104014

M.ANUDFEEP 720819104067

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

HINDUSTHAN COLLEGE OF ENGINEERING AND TECHNOLOGY

Approved by AICTE, New Delhi, Accredited with 'A' Grade by NAAC

(An Autonomous Institution, Affiliated to Anna University, Chennai)

COIMBATORE – 641 032

November 2022

TABLE OF CONTENTS

ABSTRACT

The environmental challenges the world faces nowadays have never been greater or more complex. Global areas covered by forests and urban woodlands are threatened by natural disasters that have increased dramatically during the last decades, in terms of both frequency and magnitude. Large-scale forest fires are one of the most harmful natural hazards affecting climate change and life around the world. Thus, to minimize their impacts on people and nature, the adoption of well-planned and closely coordinated effective prevention, early warning, and response approaches are necessary. This paper presents an overview of the optical remote sensing technologies used in early fire warning systems and provides an extensive survey on both flame and smoke detection algorithms employed by each technology. Three types of systems are identified, namely terrestrial, airborne, and spaceborne-based systems, while various models aiming to detect fire occurrences with high accuracy in challenging environments are studied. Finally, the strengths and weaknesses of fire detection systems based on optical remote sensing are discussed aiming to contribute to future research projects for the development of early warning fire systems.

CHAPTER-1

INTODUCTION

Forest fires have been and still are serious problem for the European Union and for all other countries in Europe. In the year 2000, the EU has established the European Forest Fire Information system (EFFIS) , which will soon become part of the European Emergency Management Service, maintained by the Copernicus Earth Observation Programme. This system provides valuable near real-time and also historical data on the forest fires in Europe, the Middle East and North Africa. Currently EFFIS is being used and supported with data by 25 EU member states and by numerous other countries. According to the annual report of EFFIS for 2016, more than 54 000 forest fires have occurred all around Europe and they have led to nearly 376 thousand hectares of burnt areas. If we compare these values to the average values from the EFFIS reports for the period 2006-2015, the number of forest fires have decreased by 13327 or by nearly 20%.

The most important factors in the fight against the forest fires include the earliest possible detection of the fire event, the proper categorisation of the fire

and fast response from the fire services. Several different types of forest fires.

CHAPTER - 2

OBJECTIVE

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.

CHAPTER-3

IDEATION PHASE 3.1

Literature Survey Problem statement 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 Literature survey TREADITIONAL METHODS Forest fire detection and prevention are real problems faced by a number of countries. Different methods have been stated for monitoring the emergence of fires. A. Watch Towers In earlier days, the forest fires were detected by manual observations with watch towers installed in the isolated areas of forest. Though this method was accurate, it was not preferred due to manual restrictions. B. Satellite Based Systems Earth orbiting satellites have been used for detection of forest fires. Unfortunately, these satellites can provide the images of regions of the earth's surface every two days which is a very long time for fire scanning. Also the weather conditions can affect the quality of satellite images. C. Optical Sensors and Digital Camera The use of optical sensors only provides a line of sight vision, where the vision can be blocked by high trees or hills. The Camera surveillance systems were also inefficient for forest fire detection because of short distance ranges. D. Wireless Sensor Networks The sensors sense physical as well as chemical parameters. The sensors can operate in a self-healing and selforganizing wireless networking environment. The major problem with this system is that there are high

chances of false alarms due to lack of proper processing of the sensor data. In this paper, we propose a method which processes the sensor data to predict fire accurately. The sensor nodes are provided with WiFi devices and tested on grassy areas to sense temperature, humidity, pressure and various other physical parameters and send this data back to the base station. At the base station, the data is processed by a machine learning agent to give alarm

3.2 Empathy Map

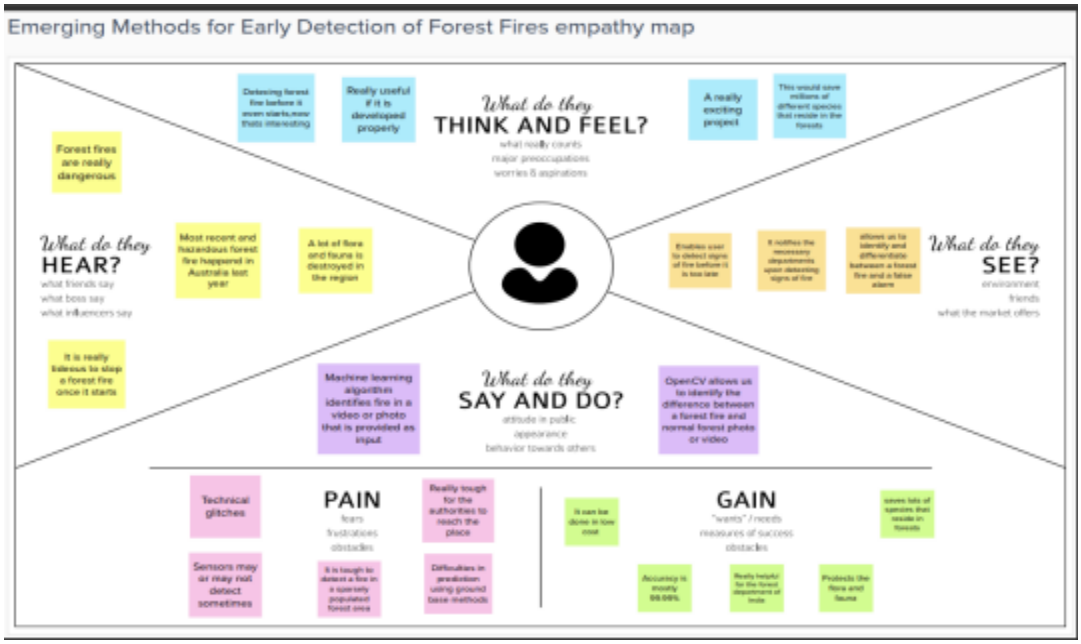


Figure 1: Empathy Map

3.3 Ideation Brainstorm & Idea Prioritization: Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

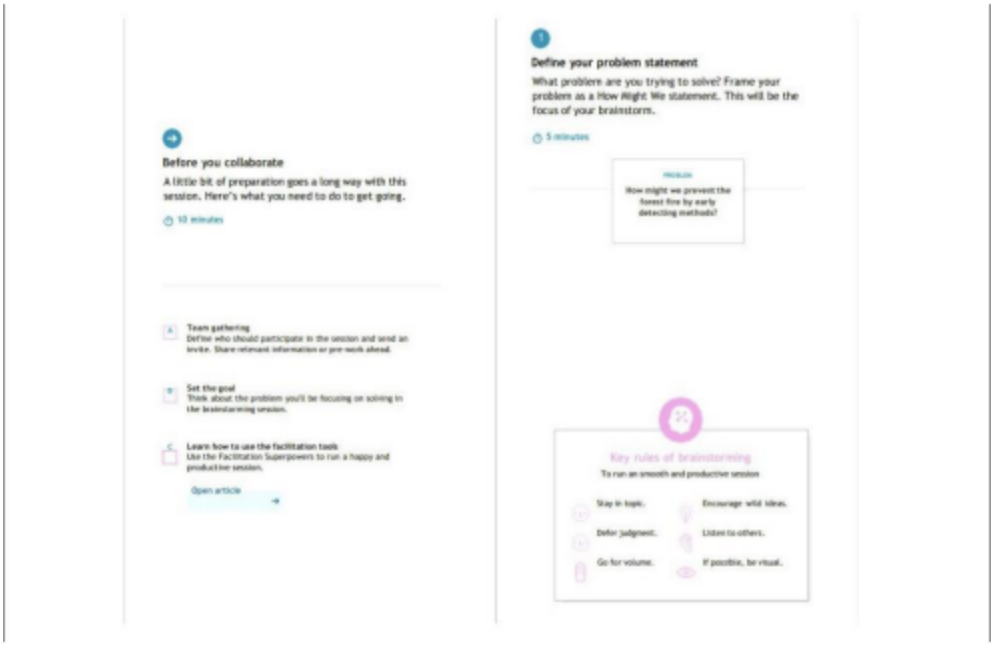


Figure 2: Ideation

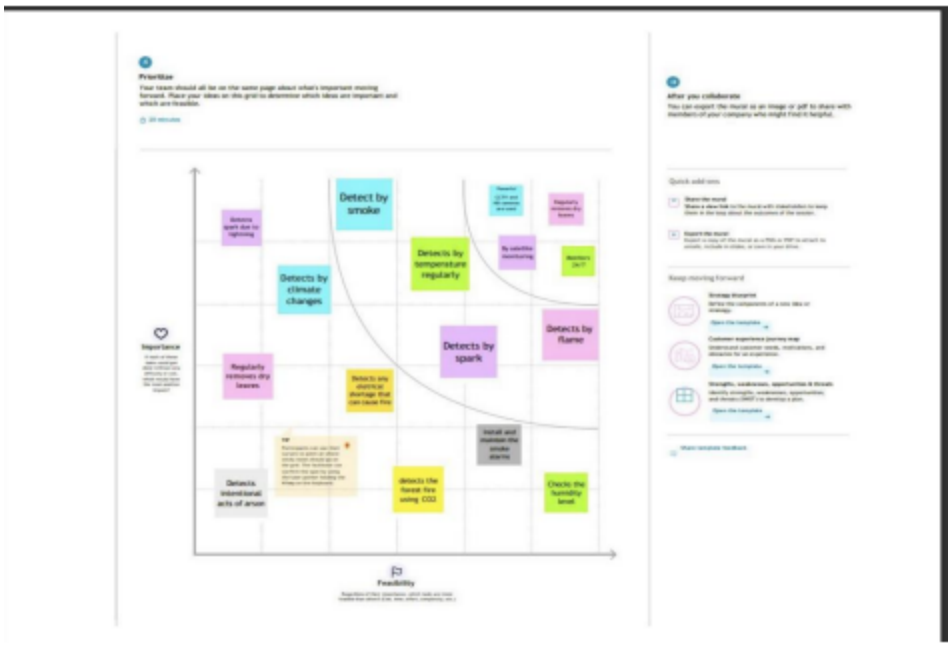
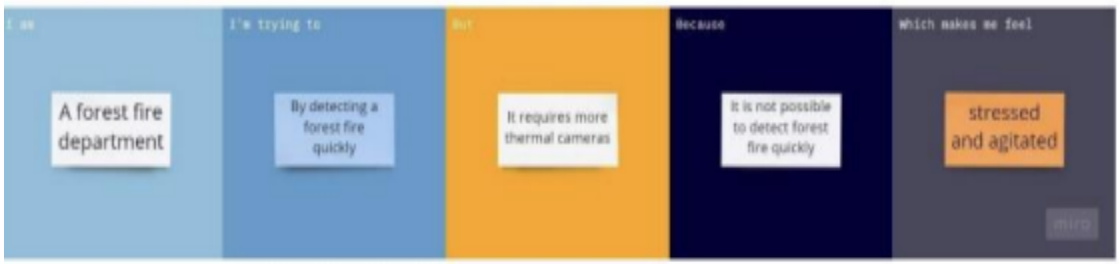


Figure 3: Ideation (Continuation)

3.4 Problem Statement



Problem Statement

Problem Statement (PS):	A Large destructive fire that spread over a forest or area of woodland is a Forest fire that causes loss of humungous amount of Property, Wildlife, Ecosystem and Economy. The project is focused on creating a permanent solution for this problem. It consists of an integrated IoT based system to detect, monitor and solve the issue without any manual involvement. The system consists of regular monitoring of the forest area with the help of cloud computing and analysis of the root cause of the fire. The system uses the latest Microcontroller, Wi-Fi communication and precision sensors such that there is no error in this part. The system also provides a quick response system so the fire can be controlled at the earliest stage.
I am (USER)	A Forest fire department
I'm trying to	Frequently monitor fire and make sure to prevent them from getting destroyed .Analyze data from various thermal camera's.
But	Requires a lot of thermal cameras for monitoring
Because	It's really hard to cover large boundaries and monitor them 24 hours a day
Which makes me feel	Stressed and agitated about the forests are burning fastly.

CHAPTER-4

PROJECT DESIGN PHASE I

4.1 Proposed Solution

S.No.	Parameter	Description
1.	Proposed Statement (Problem to be solved)	**Statement:** To find emerging methods for early detection of forest fires using artificial intelligence. **Description:** This technology is to be implemented to locate a forest or a bush fire based on the concept of deep learning and YOLO algorithm. After detecting, authorities are to be alerted immediately to mitigate any damage
2.	Idea / Solution Description	1. In case of forest fire detection the burning substances are primarily identified as sceptical flame regions using a division strategy to expel the non-fire structures and results are verified by a deep learning model. 2. The technology used to locate a forest or a bush fire is based on the concept of deep learning and YOLO algorithm. This deep learning model is deployed on a UAV which help in detection of fire, meanwhile it can be monitored by web application in order to prevent it at advance
3.	Novelty / Uniqueness	1. Accurate and reliable recognition of sceptical flame regions by means of using YOLO v3 algorithm. 2. Unlike previous algorithms, the exact location of the origin of the

		forest fire is also detected and sent to the web-app
4.	Social Impact / Customer Satisfaction	1. Because of earlier prediction, loses of life, destruction of various environmental, geographical and essential resources can be avoided. 2. By detecting a fire quickly and accurately, this system can limit the emission of toxic products created by combustion, as well as globalwarming gases produced by the fire itself
5.	Business Model (Revenue Model)	1. The software platform to provide the fully autonomous processing of data received from the camera of UAV to obtain live feed in web App. 2. This can also be implemented as a mobile application where the services can be accessed on subscription basis
6.	Scalability of the solution	1. This application can be developed as the world wide surveillance system to monitor the several sections of different forests.. 2. Filtration of false positive result by comparing the dataset with the video feed obtained.

4.2 PROBLEM SOLUTION FIT

Figure 4: Problem Solution Fit

4.3 Solution Architecture

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to: • Find the best tech solution to solve existing business problems. • Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders. • Define features, development phases, and solution requirements. • Provide specifications according to which the solution is defined, managed, and delivere

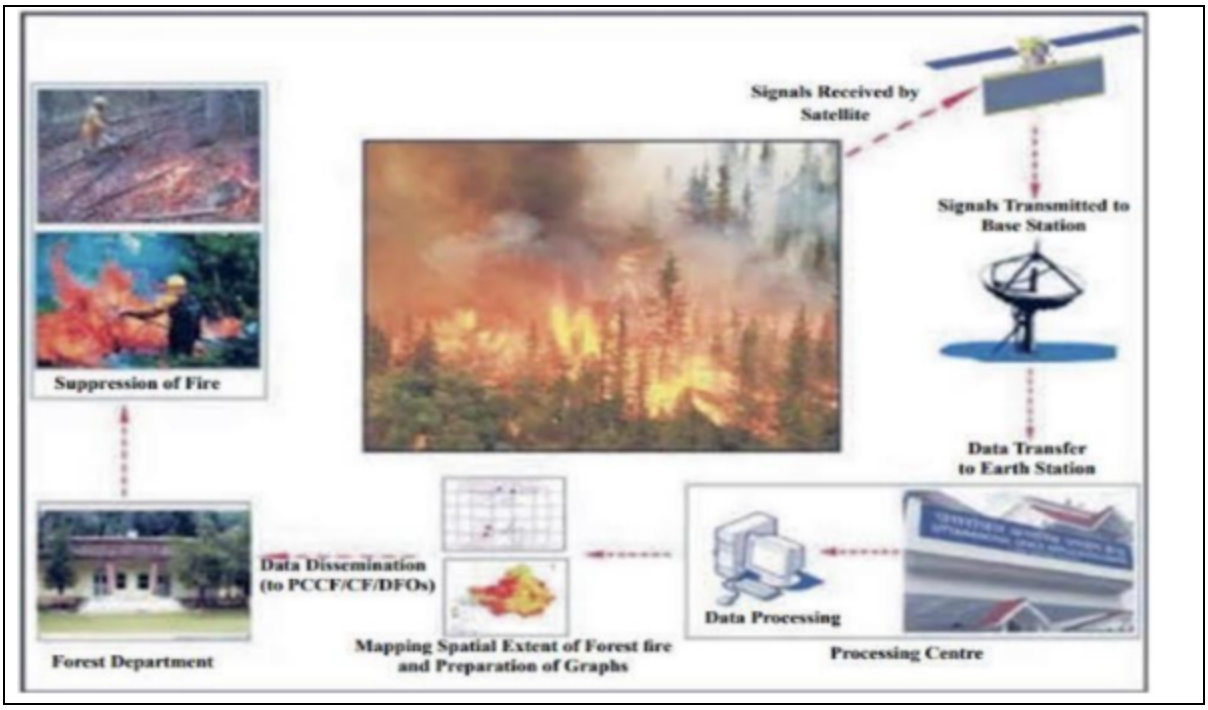


Figure 5: Solution Architecture

CHAPTER-5

PROJECT DESIGN PHASE II

5.1 Customer Journey Map

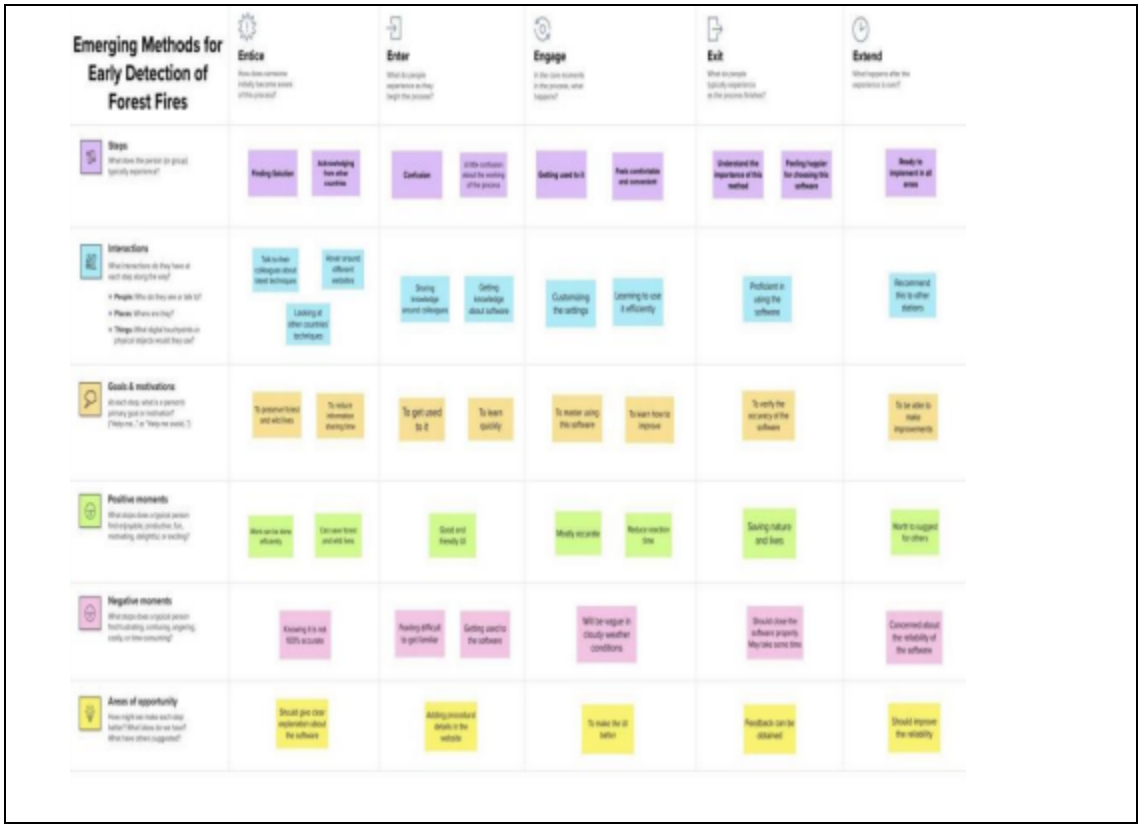


Figure 6: Customer Journey Map

5.2 SOLUTION REQUIREMENTS

Table : Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Reigstration	Registtation through Gmail.
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login using credentials
FR-4	User Search	Search for info on forest fire occurence
FR-5	User Profile	User shall be given a live feed of the forest
FR-6	User Application	User is alerted if there is an forest fire occurrence in their suuroundings

Table 4: Non-functional Requirements:

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

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Alerts according to the user location
NFR-2	**Security**	Instant live feed with alert of the siituation
NFR-3	**Reliability**	The predictions of the forest fire is 87% accurate
NFR-4	**Performance**	The feed and the alert message is an immediate action without a lag
NFR-5	**Availability**	The application gives alerts and live feeds 24/7
NFR-6	**Scalability**	Early detection and alerting users are done efficiently and in a faster means

5.3 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Example:

It is difficult to predict and detect Forest Fire in a sparsely populated forest area.

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.

If the fire is not detected, it will send the result to the frame camera. If the forest fire will detected the alert will go to the video feed frame camera

DFD:

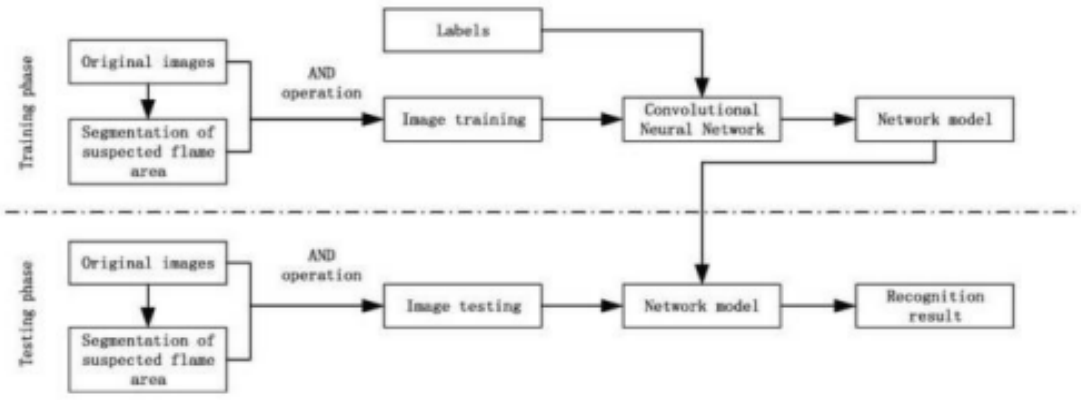


Figure 7: Data Flow Diagram

5.4 Technology Stack

Technical Architecture:

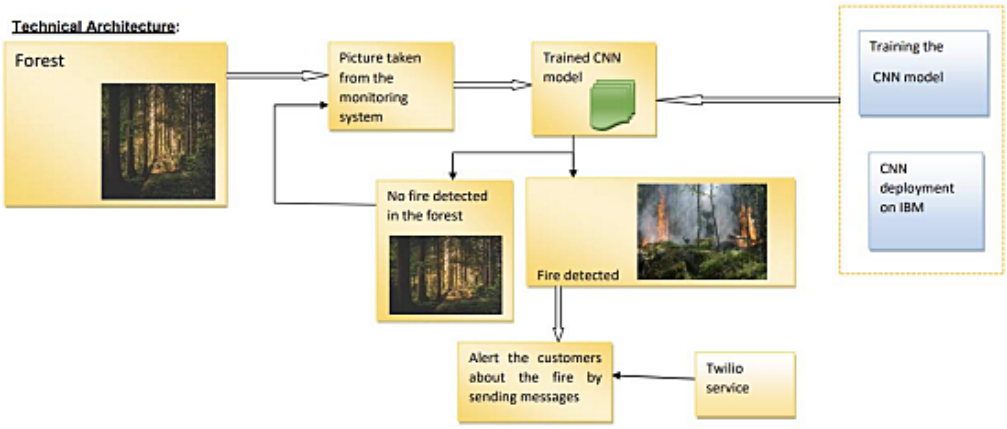


Figure 8: Technical Architecture

Table: Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	The user interacts with the application.	Python
2.	Application Logic	The logic for performance of the process to execute the desired output	Python
3.	Database	(Pictures) Composite Data Types	MySQL
4.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, CNN.
5.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud	Local, IBM cloud

Table: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	OSINT framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	OWSAP top10, SIEM

S.No	Characteristics	Description	Technology
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	HTTP overview implementation
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Round robin load balancing
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Secure cookie implementation

CHAPTER-6

PROJECT PLANNING PHASE

6.1 Milestone and Activity List

Table: Milestone & Activity List

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Gather/collect the relevant information on project use case, refer the existing solutions, technical papers, research publications etc	17 SEPTEMBER 2022
Prepare Empathy Map	Prepare the empathy map canvas to capture the user pains and gains, Prepare list of problem statements	17 SEPTEMBER 2022
Ideation	List them by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance	17 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc	19 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	19 SEPTEMBER 2022
Solution Architecture	Prepare solution architecture document.	19 SEPTEMBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	03 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document.	03 OCTOBER 2022
Data Flow Diagram	Prepare the data flow diagrams and submit for review.	03 OCTOBER 2022
Technology Architecture	Draw the technology architecture diagram.	04 OCTOBER 2022
Prepare Milestones & Activity List	Prepare the milestones & activity list of the project.	21 OCTOBER 2022
Project Delivery Of Sprint – 1,2,3&4	Develop & submit the developed code by testing it.	IN PROGRESS

6.2 Sprint Delivery Plan

Table: Product Backlog, Sprint Schedule, and Estimation

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	User input	USN-1	As a user, I can input the particular URL in the required field and waiting for validation.	2	High	Guna,Mahesh, Yahoushuva,Chethan
Sprint-1	Feature extraction	USN-1	Here system can extract feature using heuristic and visual similarity approach	1	High	Guna,Mahesh, Yahoushuva,Chethan
Sprint-1	Prediction	USN-1	Here the Model will predict the URL websites using Machine Learning algorithms	2	High	Guna,Mahesh, Yahoushuva,Chethan
Sprint-1	Classifier	USN-1	Here it will send all the model output to classifier in order to produce final result	2	High	Guna,Mahesh, Yahoushuva,Chethan
Sprint-1	Announcement	USN-1	Displays whether website is a legal site or a phishing site.	1	High	Guna,Mahesh, Yahoushuva,Chethan
Sprint-2	Bugs	USN-2	As a user, I can report bugs in the application	1	Medium	Guna,Mahesh, Yahoushuva,Chethan
Sprint-2	Feedback	USN-3	As a user, I can send feedback about the application and opinions for improvement	1	Low	Guna,Mahesh, Yahoushuva,Chethan
Sprint-3	Tips	USN-4	Here cyber security tips are provided for the Customers/Users	1	Low	Guna,Mahesh, Yahoushuva,Chethan

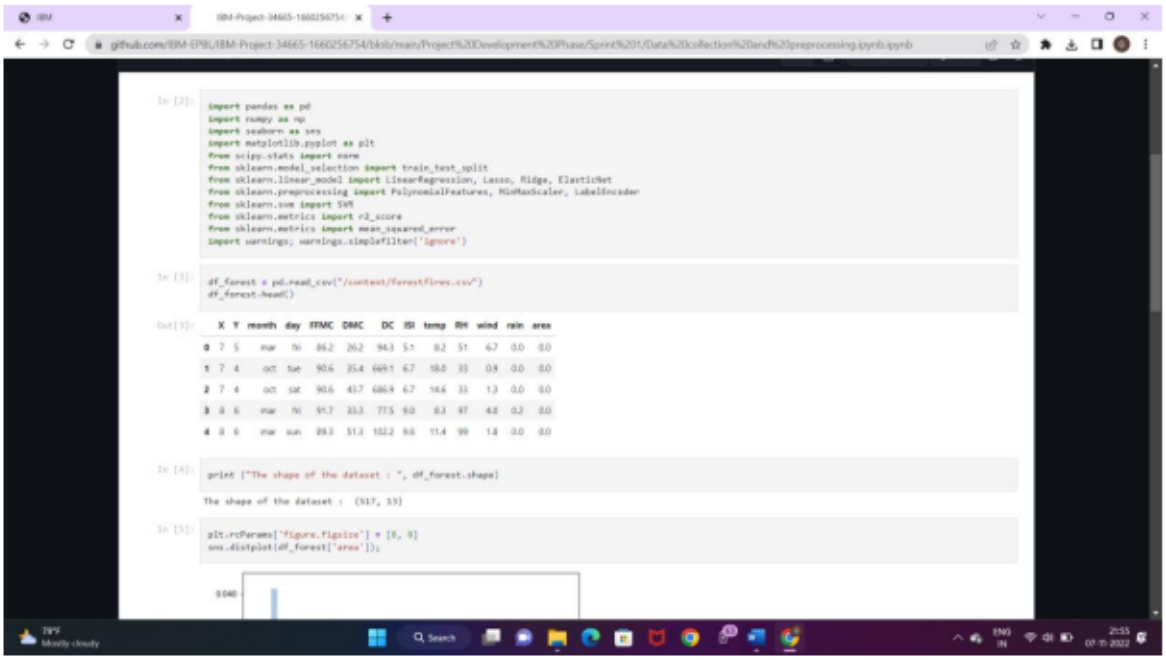
Table: Project Tracker, Velocity :

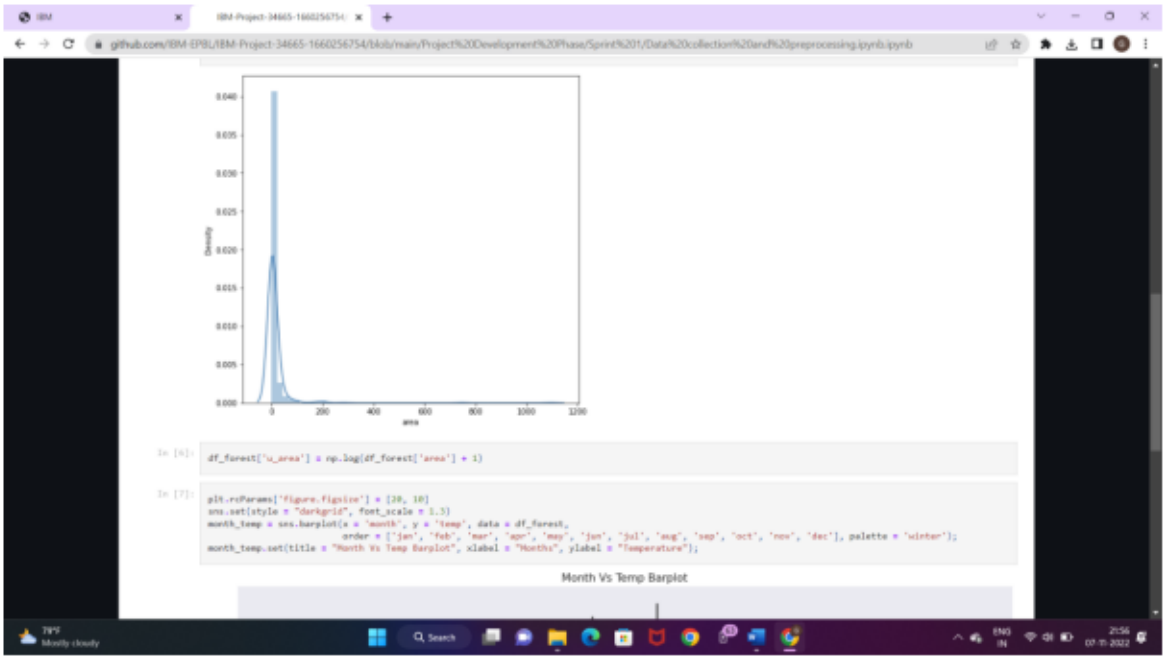
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

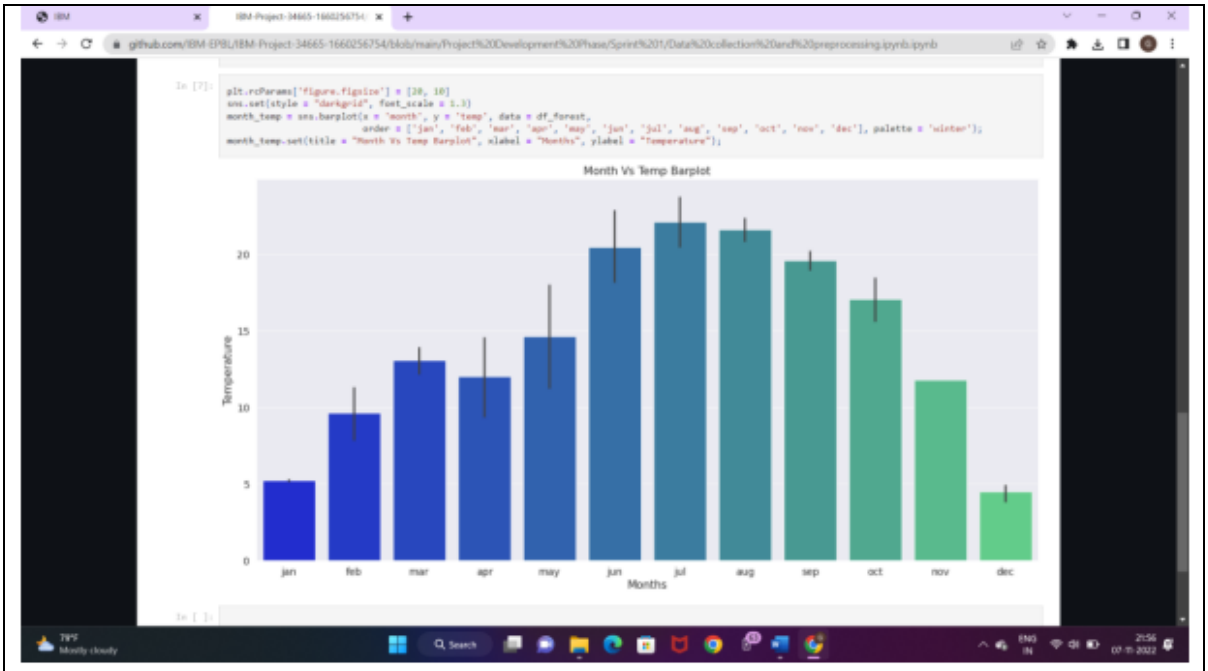
CHAPTER-7

PROJECT DEVELOPMENT PHASE

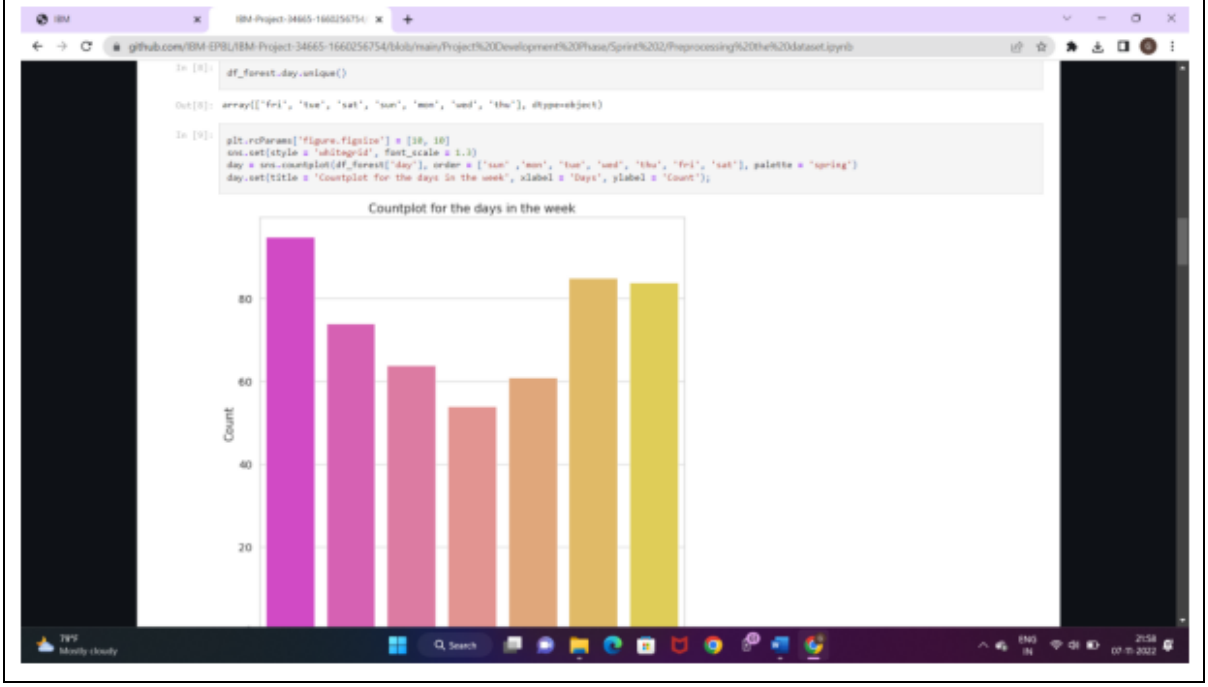
7.1 Project Development – Delivery of Sprint 1(Data collection and prepocessing.ipynb

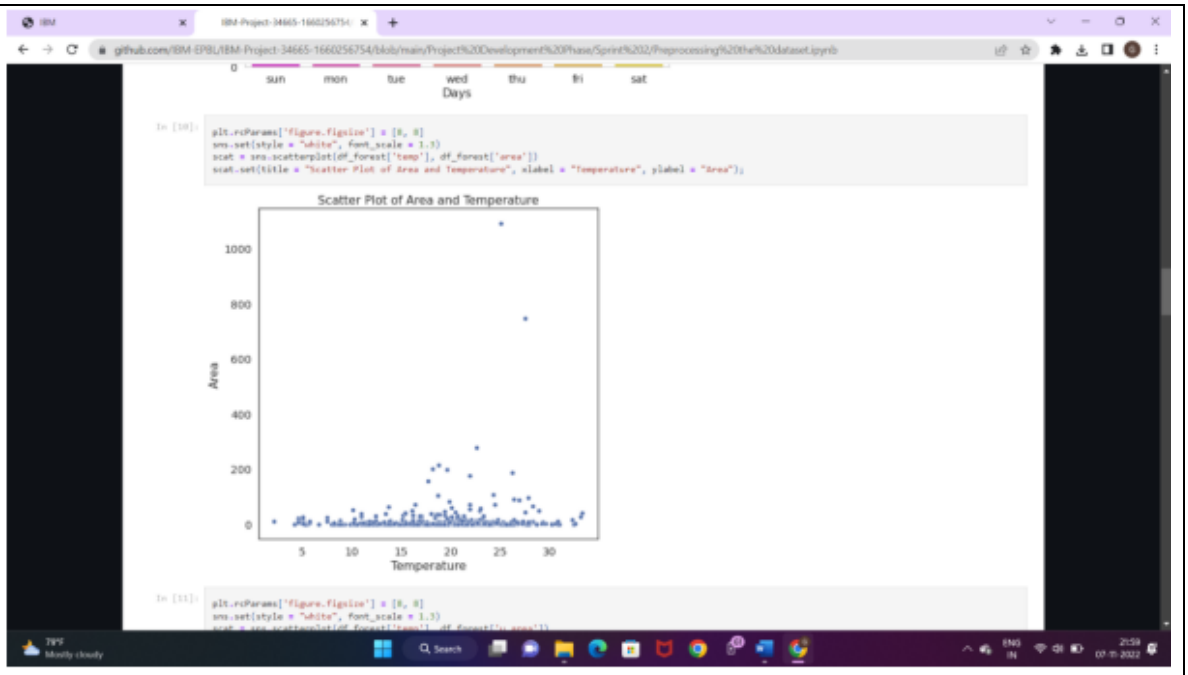


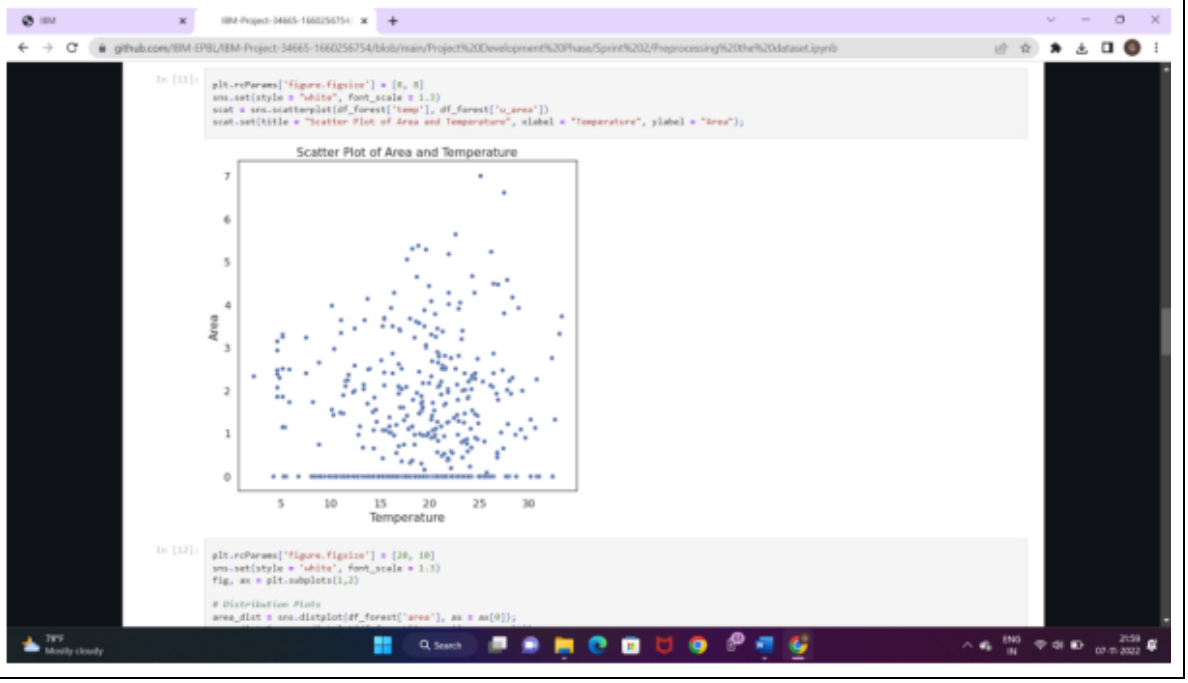


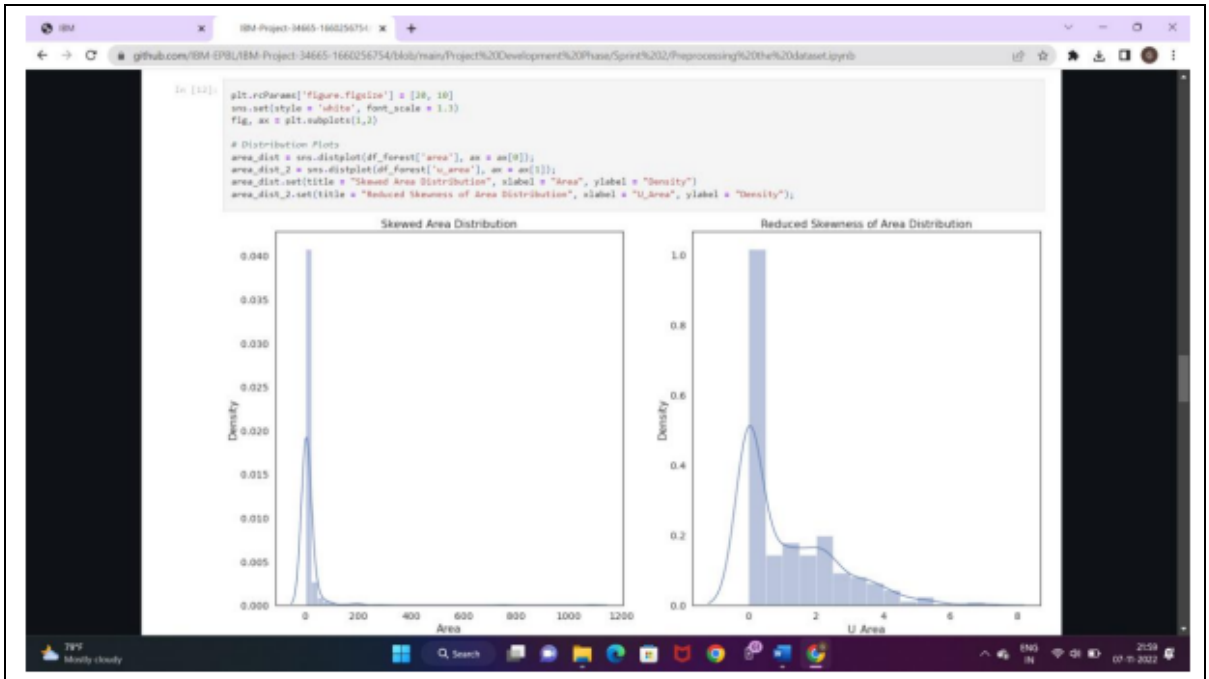


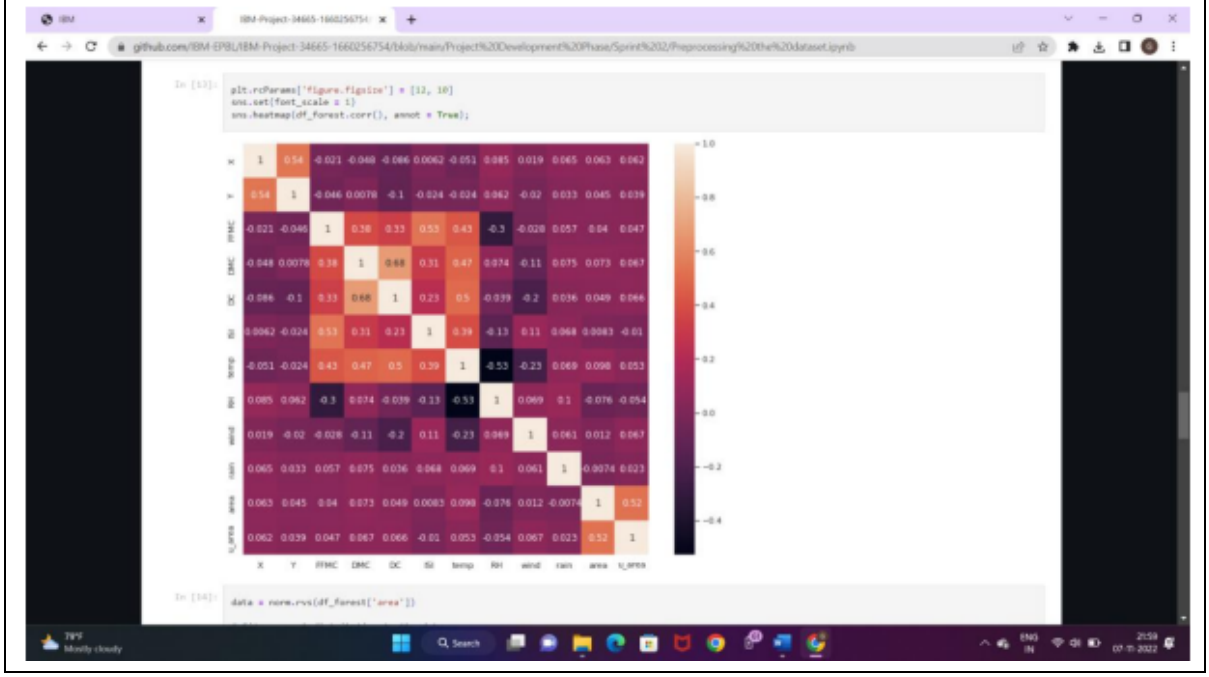
7.2 Project Development – Delivery of Sprint 2 (Preprocessing the dataset.ipynb)

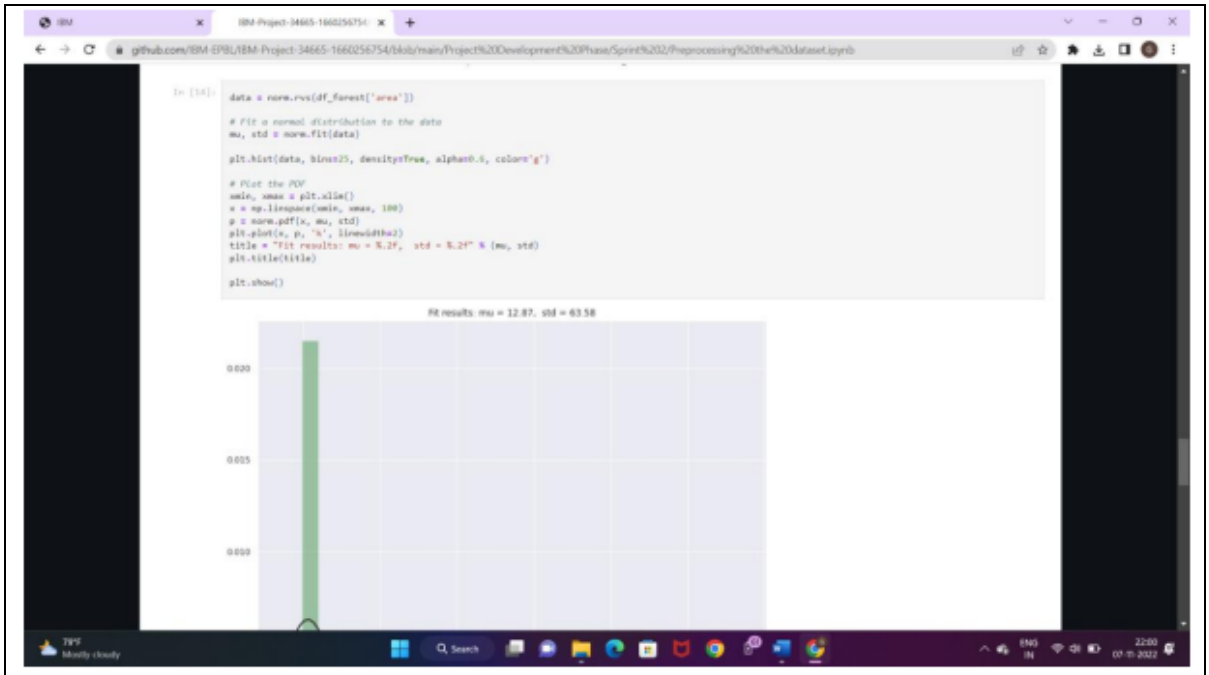


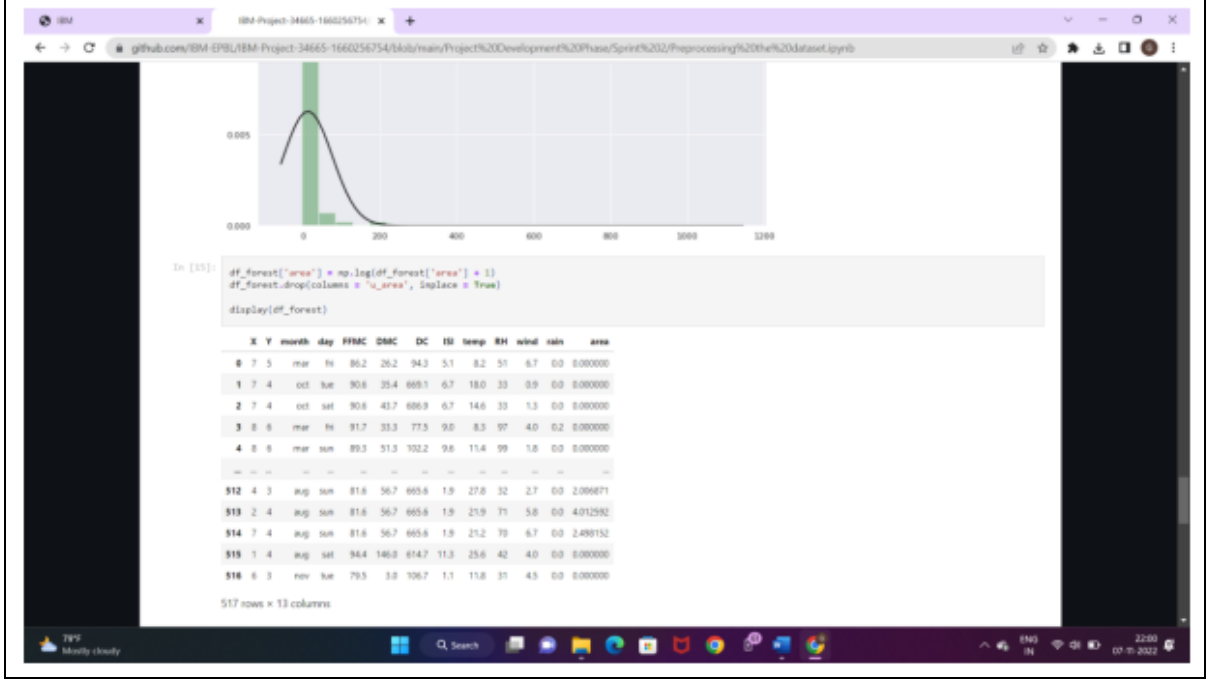

sun
mon
tue
wed
thu
fri
sat
Days
Scatter Plot of Area and Temperature
Area
Temperature

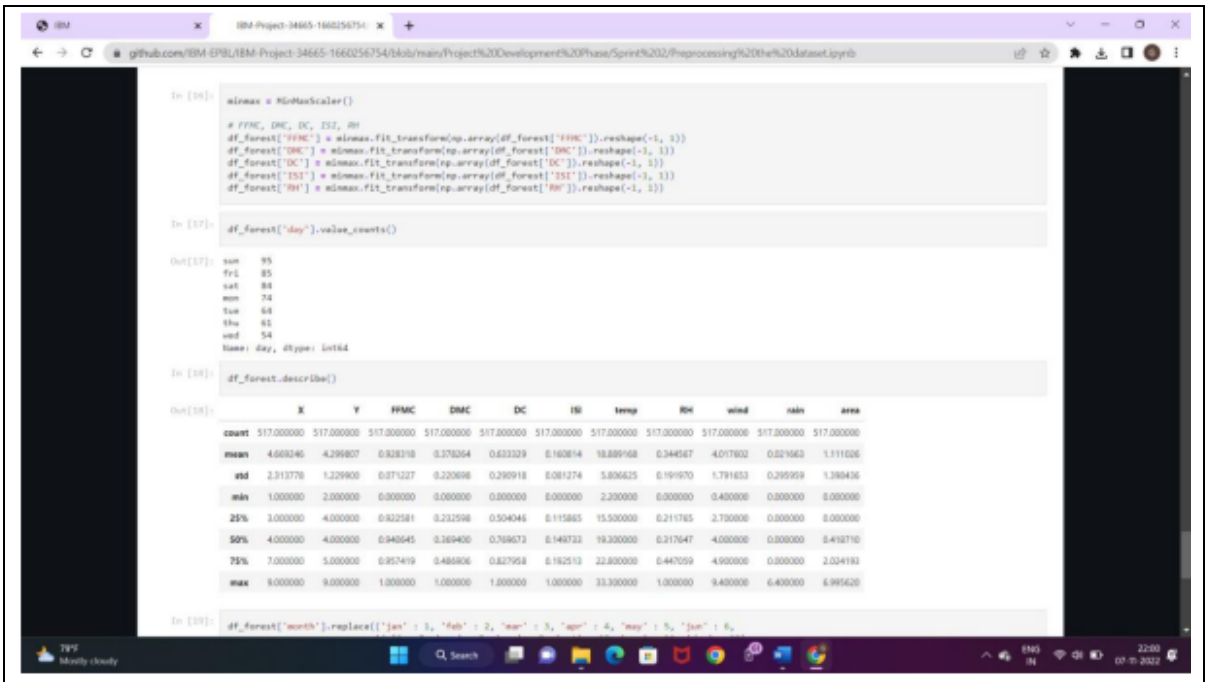

Scatter Plot of Area and Temperature
Area
Temperature


Skewed Area Distribution
Reduced Skewness of Area Distribution
Density
Area
U Area




In [34]:
data = norm.rvs(df_forest['area'])
# Fit a normal distribution to the data
mu, std = norm.fit(data)
plt.hist(data, bins=25, density=True, alpha=0.6, color='g')
# Plot the PDF
xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
p = norm.pdf(x, mu, std)
plt.plot(x, p, 'k', linewidth=2)
title = "Fit results: mu = %.2f, std = %.2f" % (mu, std)
plt.title(title)
plt.show()
Fit results: mu = 12.87, std = 63.58
0.020
0.015
0.010


0.005
0.000
0
200
400
600
800
1000
1200
In [35]:
df_forest['area'] = np.log(df_forest['area'] + 1)
df_forest.drop(columns = 'u_area', inplace = True)
display(df_forest)
X Y month day FFMC DMC DC ISI temp RH wind rain area
0 7 5 mar fri 86.2 26.2 94.3 5.1 8.2 51 6.7 0.0 0.000000
1 7 4 oct tue 90.6 35.4 669.1 6.7 18.0 33 0.9 0.0 0.000000
2 7 4 oct sat 90.6 43.7 686.9 6.7 14.6 33 1.3 0.0 0.000000
3 8 6 mar fri 91.7 33.3 77.5 9.0 8.3 97 4.0 0.2 0.000000
4 8 6 mar sun 89.3 51.3 102.2 9.6 11.4 99 1.8 0.0 0.000000
512 4 3 aug sun 81.6 56.7 665.6 1.9 27.8 32 2.7 0.0 2.006871
513 2 4 aug sun 81.6 56.7 665.6 1.9 21.9 71 5.8 0.0 4.012592
514 7 4 aug sun 81.6 56.7 665.6 1.9 21.2 70 6.7 0.0 2.498152
515 1 4 aug sat 94.4 146.0 614.7 11.3 25.6 42 4.0 0.0 0.000000
516 6 3 nov tue 79.5 3.0 106.7 1.1 11.8 31 4.5 0.0 0.000000
517 rows × 13 columns



```
In [16]:
minmax = MinMaxScaler()

# FFMC, DMC, DC, ISI, RH
df_forest['FFMC'] = minmax.fit_transform(np.array(df_forest['FFMC']).reshape(-1, 1))
df_forest['DMC'] = minmax.fit_transform(np.array(df_forest['DMC']).reshape(-1, 1))
df_forest['DC'] = minmax.fit_transform(np.array(df_forest['DC']).reshape(-1, 1))
df_forest['ISI'] = minmax.fit_transform(np.array(df_forest['ISI']).reshape(-1, 1))
df_forest['RH'] = minmax.fit_transform(np.array(df_forest['RH']).reshape(-1, 1))
```

```
In [17]:
df_forest['day'].value_counts()
```

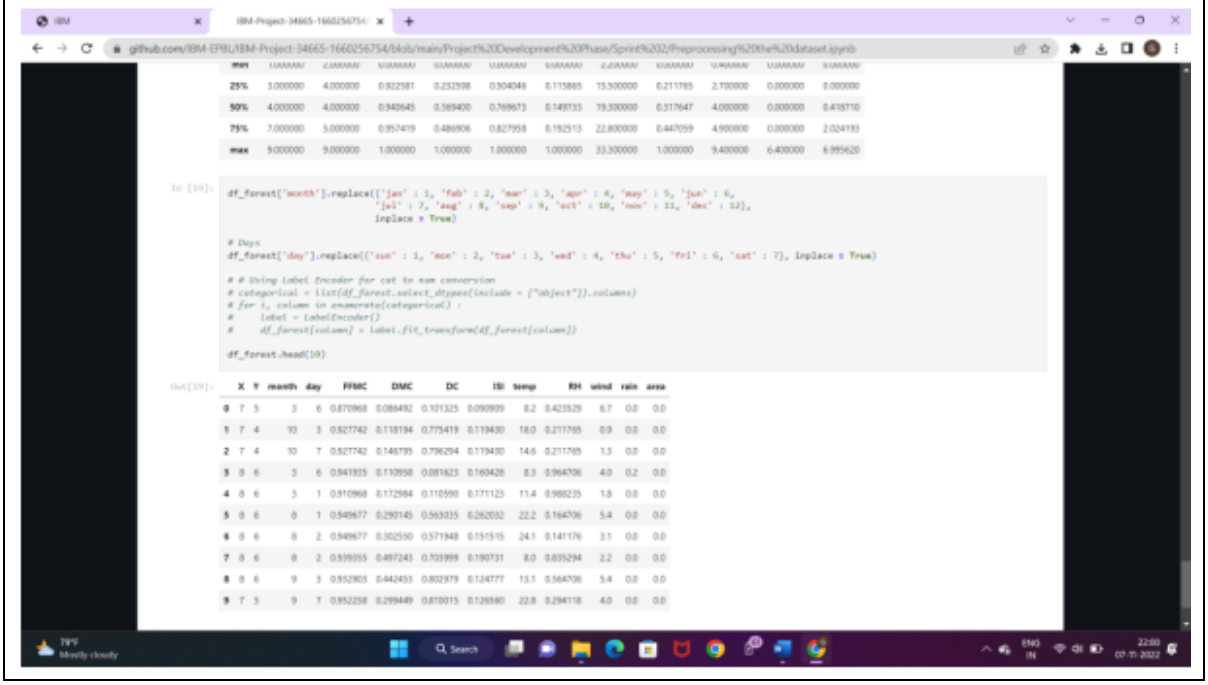
```
Out[17]:
sun    95
fri    85
sat    84
mon    74
tue    64
thu    61
wed    54
Name: day, dtype: int64
```

```
In [18]:
df_forest.describe()
```

Out[18]:

	X	Y	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
count	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000	517.000000
mean	4.669246	4.299807	0.928118	0.378264	0.633329	0.160814	18.889168	0.344567	4.017602	0.021663	1.111026
std	2.313778	1.229900	0.071227	0.220698	0.290918	0.081274	5.806625	0.191970	1.791653	0.295959	1.280436
min	1.000000	2.000000	0.000000	0.000000	0.000000	0.000000	2.200000	0.000000	0.400000	0.000000	0.000000
25%	3.000000	4.000000	0.922581	0.232598	0.504046	0.115865	15.500000	0.211765	2.700000	0.000000	0.000000
50%	4.000000	4.000000	0.940645	0.369400	0.769673	0.149733	19.300000	0.317647	4.000000	0.000000	0.419710
75%	7.000000	5.000000	0.957419	0.486806	0.827958	0.192513	22.800000	0.447059	4.900000	0.000000	2.024193
max	9.000000	9.000000	1.000000	1.000000	1.000000	1.000000	33.300000	1.000000	9.400000	6.400000	6.995620

```
In [19]:
df_forest['month'].replace({'jan' : 1, 'feb' : 2, 'mar' : 3, 'apr' : 4, 'may' : 5, 'jun' : 6,
```

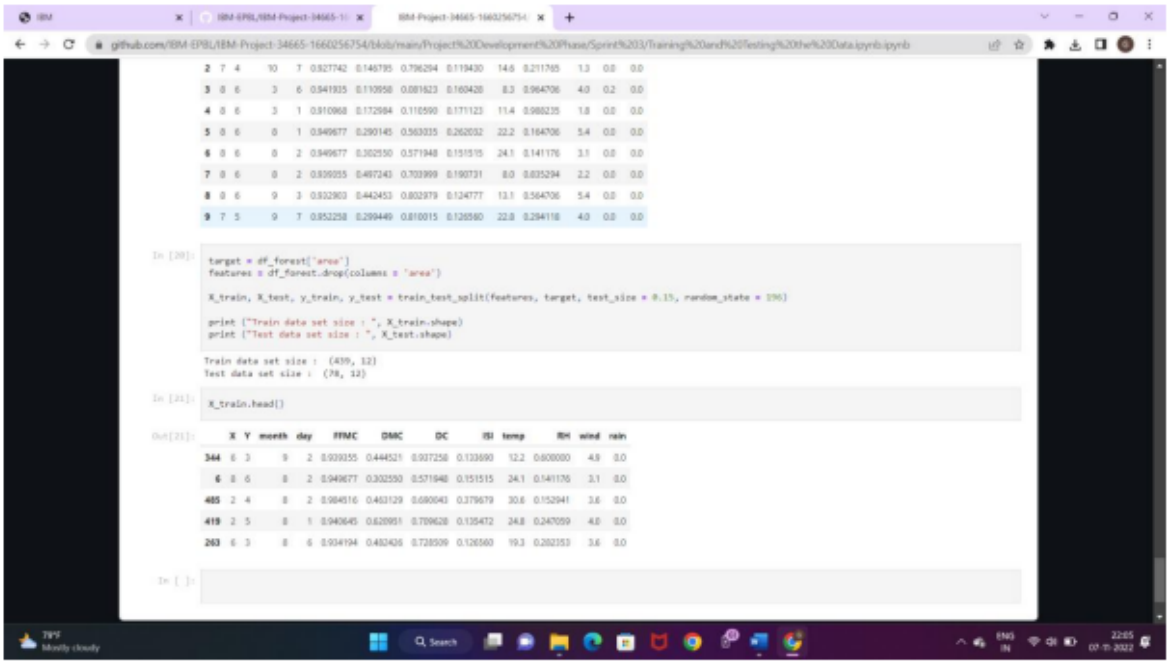


```
In [19]:
df_forest['month'].replace({'jan' : 1, 'feb' : 2, 'mar' : 3, 'apr' : 4, 'may' : 5, 'jun' : 6,
                            'jul' : 7, 'aug' : 8, 'sep' : 9, 'oct' : 10, 'nov' : 11, 'dec' : 12},
                           inplace = True)

# Days
df_forest['day'].replace({'sun' : 1, 'mon' : 2, 'tue' : 3, 'wed' : 4, 'thu' : 5, 'fri' : 6, 'sat' : 7}, inplace = True)

# # Using Label Encoder for cat to num conversion
# categorical = list(df_forest.select_dtypes(include = ["object"]).columns)
# for i, column in enumerate(categorical) :
#     label = LabelEncoder()
#     df_forest[column] = label.fit_transform(df_forest[column])

df_forest.head(10)
```

Out[19]:

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain	area
0	7	5	3	6	0.870968	0.086492	0.101325	0.090909	8.2	0.423529	6.7	0.0	0.0
1	7	4	10	3	0.927742	0.118194	0.775419	0.119430	18.0	0.211765	0.9	0.0	0.0
2	7	4	10	7	0.927742	0.146735	0.796294	0.119430	14.6	0.211765	1.3	0.0	0.0
3	8	6	3	6	0.941935	0.110958	0.081623	0.160428	8.3	0.964706	4.0	0.2	0.0
4	8	6	3	1	0.910968	0.172584	0.110590	0.171123	11.4	0.988235	1.8	0.0	0.0
5	8	6	8	1	0.949677	0.290145	0.563035	0.262032	22.2	0.164706	5.4	0.0	0.0
6	8	6	8	2	0.949677	0.302550	0.571948	0.151515	24.1	0.141176	3.1	0.0	0.0
7	8	6	8	2	0.939355	0.497243	0.703999	0.190731	8.0	0.835294	2.2	0.0	0.0
8	8	6	9	3	0.952903	0.442453	0.802979	0.124777	13.1	0.564706	5.4	0.0	0.0
9	7	5	9	7	0.952258	0.299449	0.810015	0.126560	22.8	0.294118	4.0	0.0	0.0

7.3 Project Development – Delivery of Sprint 3(Training and testing the data.ipynb)



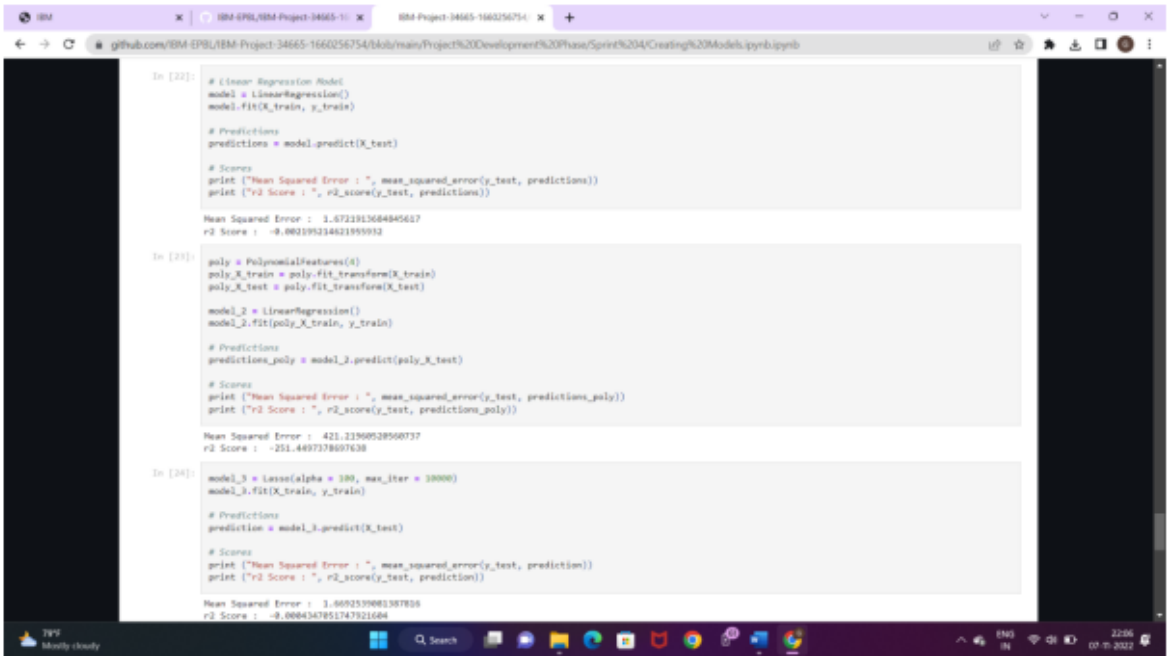
```
target = df_forest['area']
features = df_forest.drop(columns = 'area')

X_train, X_test, y_train, y_test = train_test_split(features, target, test_size = 0.15, random_state = 196)

print ("Train data set size : ", X_train.shape)
print ("Test data set size : ", X_test.shape)
```

```
Train data set size :  (439, 12)
Test data set size :  (78, 12)
```

```
X_train.head()
```

	X	Y	month	day	FFMC	DMC	DC	ISI	temp	RH	wind	rain
344	6	3		9	2	0.933055	0.444521	0.937258	0.133690	12.2	0.600000	4.9	0.0
6	8	6		8	2	0.949677	0.302550	0.571948	0.151515	24.1	0.141176	3.1	0.0
485	2	4		8	2	0.984516	0.463129	0.680043	0.379679	30.6	0.152941	3.6	0.0
419	2	5		8	1	0.940645	0.620951	0.709628	0.135472	24.8	0.247059	4.0	0.0
263	6	3		8	6	0.934194	0.482426	0.728509	0.126560	19.3	0.282353	3.6	0.0

7.4 Project Development – Delivery of Sprint 4 (Creating Models.ipynb)



```
# Linear Regression Model
model = LinearRegression()
model.fit(X_train, y_train)

# Predictions
predictions = model.predict(X_test)

# Scores
print ("Mean Squared Error : ", mean_squared_error(y_test, predictions))
print ("r2 Score : ", r2_score(y_test, predictions))
```

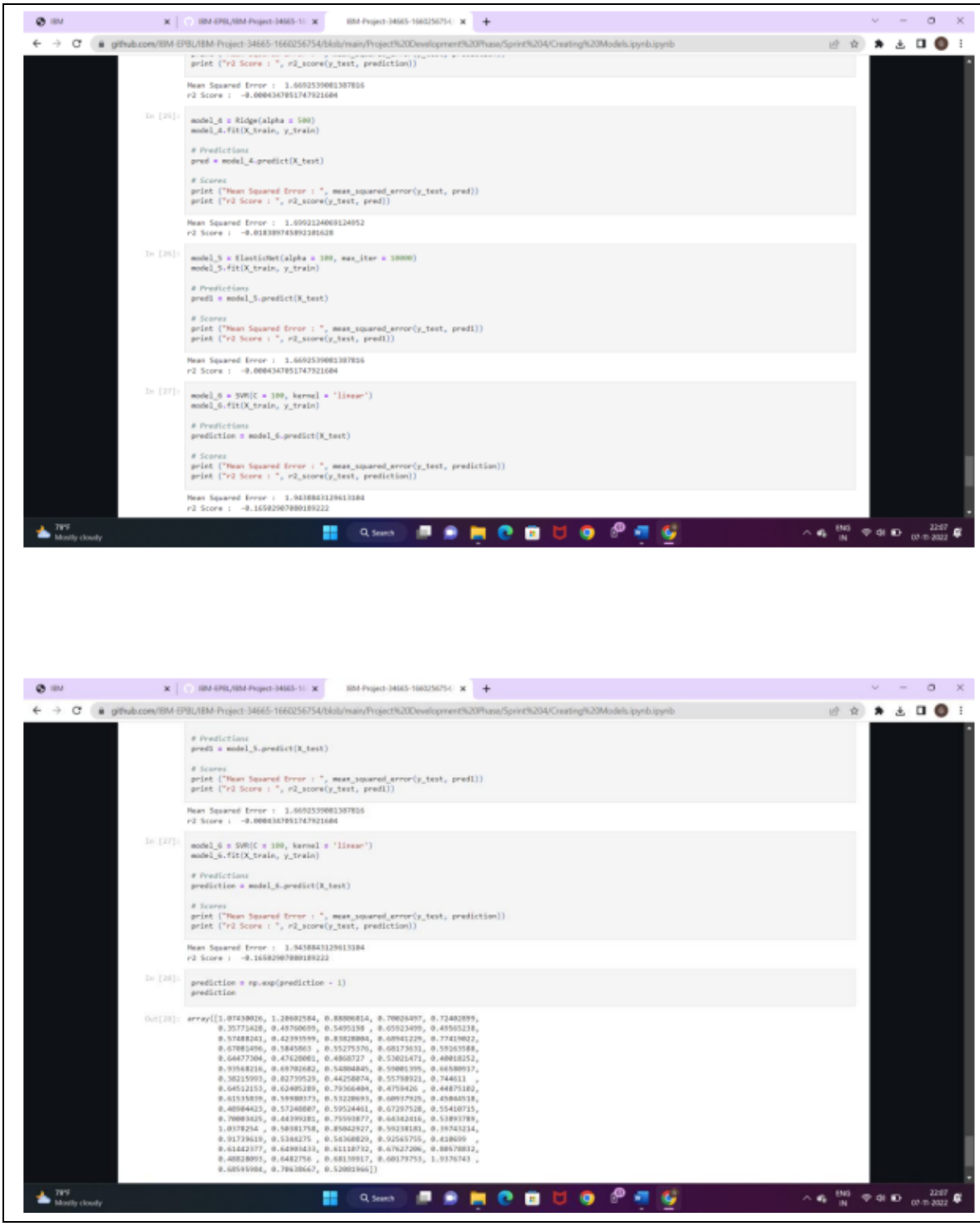
```
Mean Squared Error :  1.6721913684845617
r2 Score :  -0.0011952134621955932
```

```
poly = PolynomialFeatures(4)
poly_X_train = poly.fit_transform(X_train)
poly_X_test = poly.fit_transform(X_test)

model_2 = LinearRegression()
model_2.fit(poly_X_train, y_train)

# Predictions
predictions_poly = model_2.predict(poly_X_test)

# Scores
print ("Mean Squared Error : ", mean_squared_error(y_test, predictions_poly))
print ("r2 Score : ", r2_score(y_test, predictions_poly))
```

```
Mean Squared Error :  421.21560528560737
r2 Score :  -251.44973786937638
```

```
model_3 = Lasso(alpha = 100, max_iter = 10000)
model_3.fit(X_train, y_train)

# Predictions
prediction = model_3.predict(X_test)

# Scores
print ("Mean Squared Error : ", mean_squared_error(y_test, prediction))
print ("r2 Score : ", r2_score(y_test, prediction))
```

```
Mean Squared Error :  1.6692539981387816
r2 Score :  -0.0004347051747921604
```


print ("r2 Score : ", r2_score(y_test, prediction))
Mean Squared Error : 1.6692539081387816
r2 Score : -0.0004347851747921604
In [26]:
model_4 = Ridge(alpha = 500)
model_4.fit(X_train, y_train)
# Predictions
pred = model_4.predict(X_test)
# Scores
print ("Mean Squared Error : ", mean_squared_error(y_test, pred))
print ("r2 Score : ", r2_score(y_test, pred))
Mean Squared Error : 1.6992124069124952
r2 Score : -0.018389745892181628
In [26]:
model_5 = ElasticNet(alpha = 100, max_iter = 10000)
model_5.fit(X_train, y_train)
# Predictions
pred1 = model_5.predict(X_test)
# Scores
print ("Mean Squared Error : ", mean_squared_error(y_test, pred1))
print ("r2 Score : ", r2_score(y_test, pred1))
Mean Squared Error : 1.6692539081387816
r2 Score : -0.0004347851747921604
In [27]:
model_6 = SVR(C = 100, kernel = 'linear')
model_6.fit(X_train, y_train)
# Predictions
prediction = model_6.predict(X_test)
# Scores
print ("Mean Squared Error : ", mean_squared_error(y_test, prediction))
print ("r2 Score : ", r2_score(y_test, prediction))
Mean Squared Error : 1.9438845129613584
r2 Score : -0.16582907880189222
79°F
Mostly cloudy
Search
ENG
IN
22:07
07-11-2022

In [28]:
prediction = np.exp(prediction - 1)
prediction
Out[28]:
array([1.07430826, 1.28682584, 0.88806814, 0.70026497, 0.72402899,
0.35771428, 0.49760659, 0.5495198 , 0.65923499, 0.49565238,
0.57488241, 0.42391599, 0.83828804, 0.68941229, 0.77419022,
0.67081496, 0.5845863 , 0.55275376, 0.68173631, 0.59265588,
0.64477504, 0.47620081, 0.4868727 , 0.53021471, 0.48018152,
0.91568216, 0.69702682, 0.54804845, 0.59001395, 0.66580917,
0.58215993, 0.82739529, 0.44258074, 0.55798921, 0.744611 ,
0.64512153, 0.63405289, 0.79366404, 0.4758426 , 0.44875102,
0.61535839, 0.59980173, 0.53228695, 0.68937925, 0.45866518,
0.48984423, 0.57248607, 0.59524461, 0.67297528, 0.55410715,
0.70003625, 0.44399285, 0.75593877, 0.66362816, 0.53893789,
1.0178254 , 0.58381758, 0.85042927, 0.59238183, 0.59743314,
0.91739619, 0.5346275 , 0.56360829, 0.92565755, 0.618699 ,
0.61442377, 0.64989434, 0.61118732, 0.6762706, 0.88678832,
0.48828093, 0.6482756 , 0.68139917, 0.60179753, 1.5976743 ,
0.68595984, 0.78620667, 0.52001966])
79°F
Mostly cloudy
Search
ENG
IN
22:07
07-11-2022

CHAPTER-8

CONCLUSION

To limit the damage caused by forest fires and to control the start of fires and its spread, we have presented in this study a method of early detection of forest fires. This method is based on three steps: Estimate the general risk level of the forest, assess and predict in several places the existence or not of fires, and alert the necessary first responders to quell the spread of the fires. The originality of this work lies in the use of a wireless sensor and RF network distributed over the entire forest area and the deep learning methods to predict in real-time a possible origination and predicted path of the forest fire.

CHAPTER-9

REFERENCES

[1] Y. Huang and K. Boyle, Antennas, "From Theory to Practice.", West Sussex, United Kingdom: John Wiley & Sons Ltd, 2008.

[2] AA Portable Power Corp, "Category : Li-Ion/Polymer Single Cells," 2019. [Online]. Available: https://www.batteryspace.com/liionsinglecell.aspx.

[3] C. A. Balanis, Antenna Theory, "Analysis and Design," Fourth Edition., Hoboken, New Jersey: John Wiley and Sons, Inc., 2016.

[4] J. Pike, "Understanding LoRa WAN Basics: A Non-Technical Explanation," 21 August 2017. [Online]. Available: https://metova.com/ understanding-lora-basics-a-non-technical-explanation/.

[5] "LoRaWAN, "What is it?"," November 2015. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-04/what-is-lorawan.pdf.

CHAPTER-10
APPENDIX

GITHUBLINK

https://github.com/IBM-EPBL/IBM-Project-35327-1660283496

DEMOLINK

https://youtu.be/dW_fMttTPho