**CORE FUNCTIONALITIES OF PHASE 3**

**Deploy Flask:**

* In the module the code was deployed using flask web framework
* The prediction page runs on the designated local host server.
* The local host link will be a output of the phase 3 code

**HTML:**

<!DOCTYPE html>

<html >

<!--From https://codepen.io/frytyler/pen/EGdtg-->

<head>

<meta charset="UTF-8">

<title>Flood</title>

<link rel="stylesheet" href="{{ url\_for('static', filename='css/bootstrap.min.css') }}">

<link href='https://fonts.googleapis.com/css?family=Pacifico' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Arimo' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Hind:300' rel='stylesheet' type='text/css'>

<link href='https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300' rel='stylesheet' type='text/css'>

<style>

.back{

background-image: url("{{ url\_for('static', filename='image/img.jpg') }}");

background-size:cover;

}

.white{

color:white;

}

.space{

margin:10px 30px;

padding:15px 10px;

background: palegreen;

width:auto;

}

.nspace{

margin:10px 30px;

background: palegreen;

width:auto;

}

.gap{

padding:10px 20px;

}

</style>

</head>

<body class="back">

<div>

<div class="jumbotron">

<h1 style="text-align:center"> FLOOD PREDICTION </h1>

</div>

<!-- Main Input For Receiving Query to our ML -->

<form class="form-group" action="{{ url\_for('predict')}}"method="post">

<div class="container">

<div class="row">

<div class="gap col-md-6">

<label class="white" for="">STATE\_UT\_NAME</label>

<select class="nspace form-control" name="STATE\_UT\_NAME" id="STATE\_UT\_NAME">

<option value=0>ANDAMAN And NICOBAR ISLANDS</option>

<option value=2>ARUNACHAL PRADESH</option>

<option value=3>ASSAM</option>

<option value=22>MEGHALAYA</option>

<option value=21>MANIPUR</option>

<option value=23>MIZORAM</option>

<option value=24>NAGALAND</option>

<option value=31>TRIPURA</option>

<option value=34>WEST BENGAL</option>

<option value=29>SIKKIM</option>

<option value=25>ORISSA</option>

<option value=15>JHARKHAND</option>

<option value=4>BIHAR</option>

<option value=32>UTTAR PRADESH</option>

<option value=33>UTTARANCHAL</option>

<option value=12>HARYANA</option>

<option value=5>CHANDIGARH</option>

<option value=9>DELHI</option>

<option value=27>PUNJAB</option>

<option value=13>HIMACHAL</option>

<option value=14>JAMMU AND KASHMIR</option>

<option value=28>RAJASTHAN</option>

<option value=19>MADHYA PRADESH</option>

<option value=7>DADAR NAGAR HAVELI</option>

<option value=8>DAMAN AND DUI</option>

<option value=20>MAHARASHTRA</option>

<option value=10>GOA</option>

<option value=6>CHATISGARH</option>

<option value=1>ANDHRA PRADESH</option>

<option value=30>TAMIL NADU</option>

<option value=26>PONDICHERRY</option>

<option value=16>KARNATAKA</option>

<option value=17>KERALA</option>

<option value=18>LAKSHADWEEP</option>

</select>

<label class="white" for=""> Jan-Feb</label>

<input type="number" class="space form-control" step="0.01" name=" Jan-Feb" placeholder="Jan-Feb" required="required" /><br>

<label class="white" for="">Mar-May</label>

<input type="number" class="space form-control" step="0.01" name="Mar-May" placeholder="Mar-May" required="required" /><br>

<label class="white" for="">Jun-Sep</label>

<input type="number" class="space form-control" step="0.01" name="Jun-Sep" placeholder="Jun-Sep" required="required" /><br>

<label class="white" for="">Oct-Dec</label>

<input type="number" class="space form-control" step="0.01" name="Oct-Dec" placeholder="Oct-Dec" required="required" /><br>

<label class="white" for=""> ANNUAL</label>

<input type="number" class="space form-control" step="0.01" name="ANNUAL" placeholder="ANNUAL" required="required" /><br>

</div>

</div>

</div>

<div style="padding:2% 35%">

<button type="submit" class="btn btn-success btn-block" style="width:350px;padding:20px">Predict</button>

</div>

</form>

<br>

<br>

<div style="background:skyblue;padding:2% 40%">

{{ prediction\_text }}

</div>

</div>

</body>

</html>

**Deploy:**

import numpy as np

from flask import Flask, request, jsonify, render\_template

import pickle

import joblib

app = Flask(\_\_name\_\_)

model = joblib.load('flood.pkl')

@app.route('/')

def home():

return render\_template('index.html')

@app.route('/predict',methods=['POST'])

def predict():

'''

For rendering results on HTML GUI

'''

int\_features = [(x) for x in request.form.values()]

final\_features = [np.array(int\_features)]

print(final\_features)

prediction = model.predict(final\_features)

output = prediction [0]

print(output)

if output == 0:

output ='Not Happen'

elif output == 1:

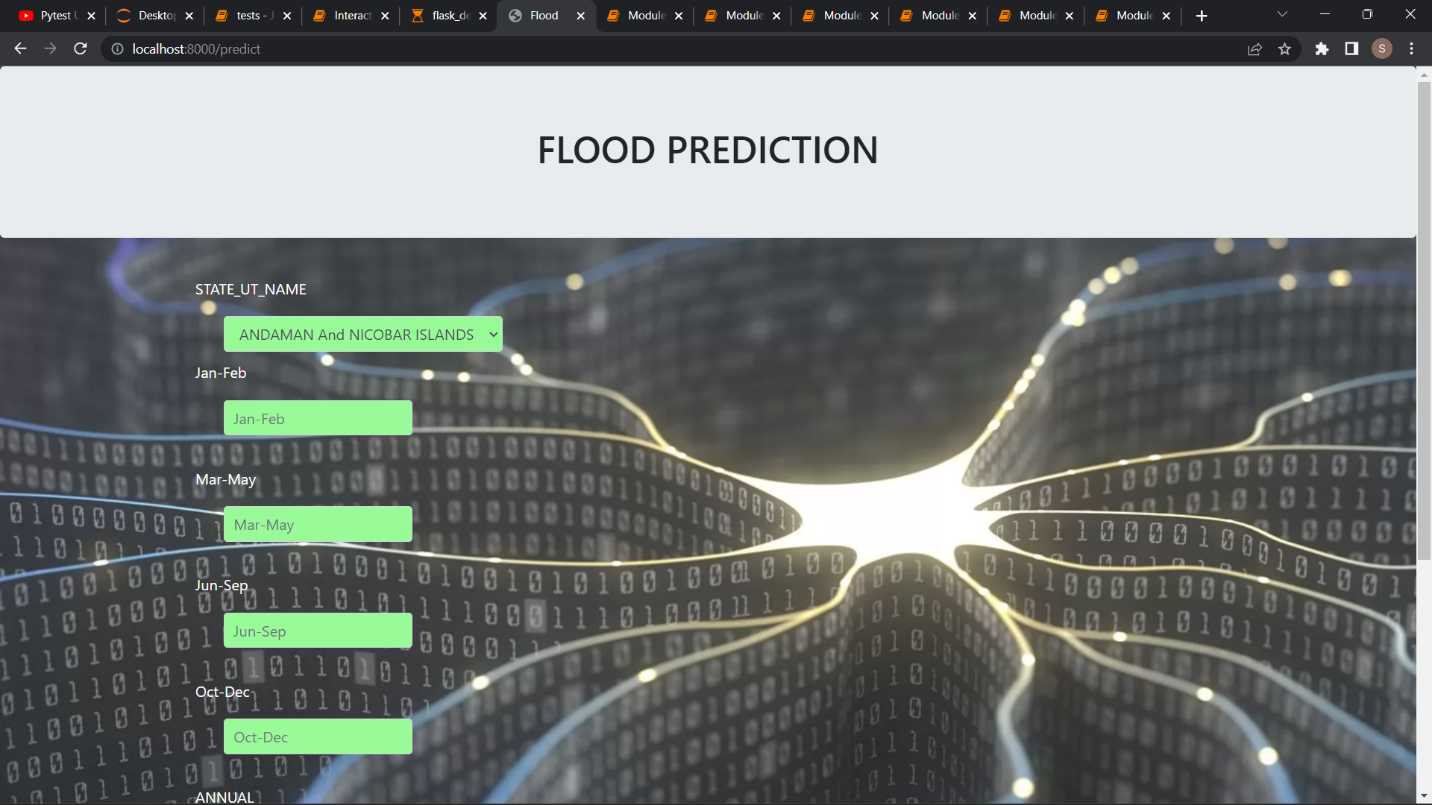
output ='Happen'

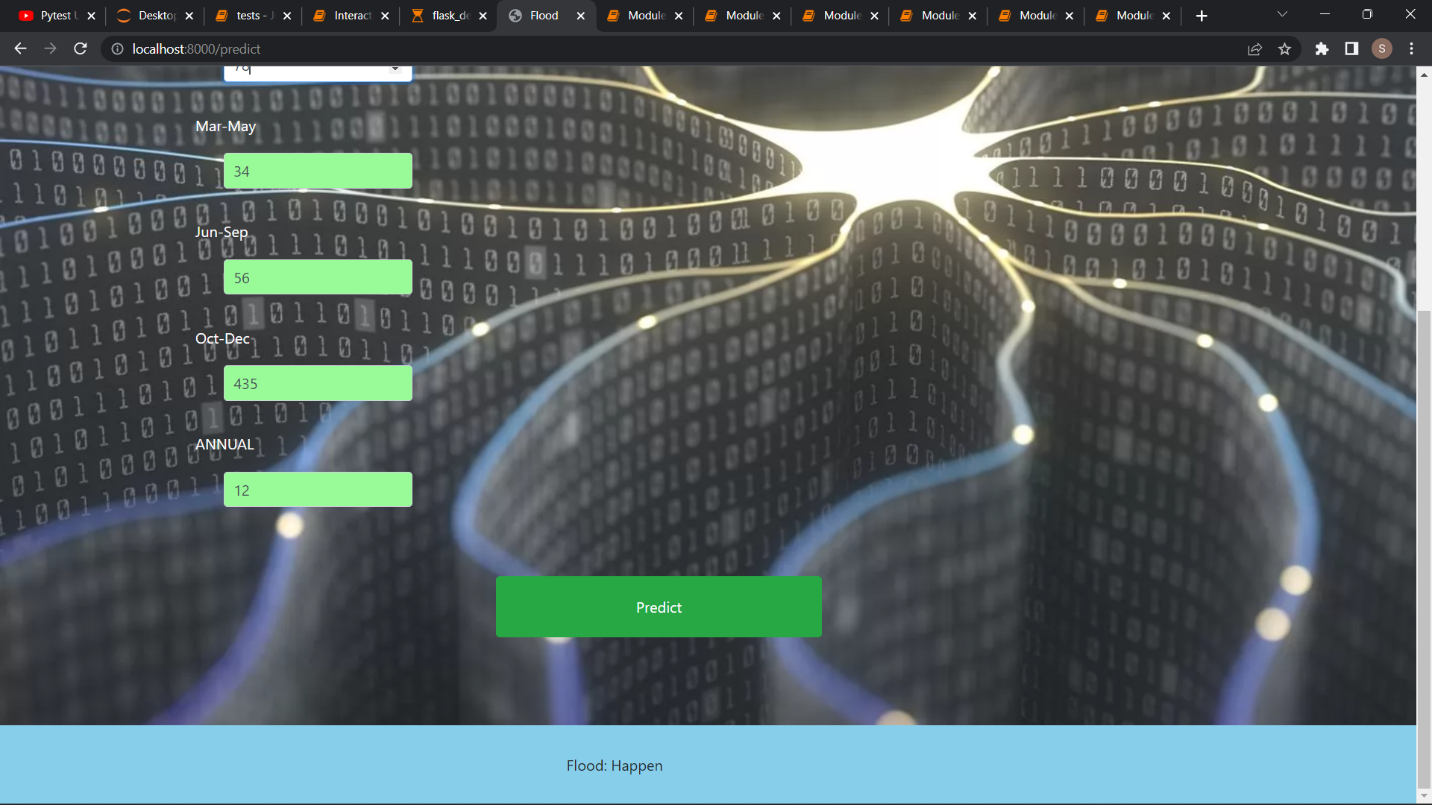
return render\_template('index.html', prediction\_text='Flood: {}'.format(output))

if \_\_name\_\_ == "\_\_main\_\_":

app.run(host="localhost", port=8000)

**Output Screenshots:**

****

****

**Dashboard Code:**

#!pip install panel

#!pip install hvplot

import pandas as pd

import numpy as np

import panel as pn

pn.extension('tabulator'**)**

import hvplot.pandas

df = pd.read\_csv('demo2.csv')

df.columns

#FIll NAs

df = df.fillna(0)

idf = df.interactive()

# Define Panel widgets

year\_slider = pn.widgets.IntSlider(name='Year Slider', start=1901, end=2015, step=5, value=1901)

year\_slider

# Radio Buttons for flood averages

yaxis\_JanFeb = pn.widgets.RadioButtonGroup(

name = 'Y axis',

options=['Jan-Feb', 'ANNUAL',],

button\_type='success'

)

states = ['ANDAMAN & NICOBAR ISLANDS','ARUNACHAL PRADESH','ASSAM & MEGHALAYA','NAGA MANI MIZO TRIPURA','SUB HIMALAYAN WEST BENGAL & SIKKIM','GANGETIC WEST BENGAL','ORISSA','JHARKHAND','BIHAR',

'WEST UTTAR PRADESH','UTTARAKHAND','HARYANA DELHI & CHANDIGARH','PUNJAB','HIMACHAL PRADESH','JAMMU & KASHMIR','WEST RAJASTHAN','WEST MADHYA PRADESH','GUJARAT REGION',

'KONKAN & GOA','MADHYA MAHARASHTRA','CHHATTISGARH','COASTAL ANDHRA PRADESH','TELANGANA', 'TAMIL NADU', 'COASTAL KARNATAKA', 'KERALA', 'LAKSHADWEEP']

JFflood\_pipeline = (

idf[

(idf.YEAR <= year\_slider) &

(idf.SUBDIVISION.isin(states))

]

.groupby(['SUBDIVISION', 'YEAR'])[yaxis\_JanFeb].mean()

.to\_frame()

.reset\_index()

.sort\_values(by='YEAR')

.reset\_index(drop=True)

)

JFflood\_plot = JFflood\_pipeline.hvplot(x= 'YEAR',by='SUBDIVISION', y=yaxis\_JanFeb,line\_width=1, title="Flood By State")

JFflood\_plot

yaxis\_MarMay = pn.widgets.RadioButtonGroup(

name = 'Y axis',

options=['Mar-May', 'ANNUAL',],

button\_type='success'

)

yaxis\_MarMay = pn.widgets.RadioButtonGroup(

name = 'Y axis',

options=['Mar-May', 'ANNUAL',],

button\_type='success'

)

states = ['ANDAMAN & NICOBAR ISLANDS','ARUNACHAL PRADESH','ASSAM & MEGHALAYA','NAGA MANI MIZO TRIPURA','SUB HIMALAYAN WEST BENGAL & SIKKIM','GANGETIC WEST BENGAL','ORISSA','JHARKHAND','BIHAR',

'WEST UTTAR PRADESH','UTTARAKHAND','HARYANA DELHI & CHANDIGARH','PUNJAB','HIMACHAL PRADESH','JAMMU & KASHMIR','WEST RAJASTHAN','WEST MADHYA PRADESH','GUJARAT REGION',

'KONKAN & GOA','MADHYA MAHARASHTRA','CHHATTISGARH','COASTAL ANDHRA PRADESH','TELANGANA', 'TAMIL NADU', 'COASTAL KARNATAKA', 'KERALA', 'LAKSHADWEEP']

MMflood\_pipeline = (

idf[

(idf.YEAR <= year\_slider) &

(idf.SUBDIVISION.isin(states))

]

.groupby(['SUBDIVISION', 'YEAR'])[yaxis\_MarMay].mean()

.to\_frame()

.reset\_index()

.sort\_values(by='YEAR')

.reset\_index(drop=True)

)

MMflood\_plot = MMflood\_pipeline.hvplot(x= 'YEAR',by='SUBDIVISION', y=yaxis\_MarMay,line\_width=1, title="Flood By State for MAR TO MAY")

MMflood\_plot

yaxis\_JunSep = pn.widgets.RadioButtonGroup(

name = 'Y axis',

options=['Jun-Sep', 'ANNUAL',],

button\_type='success'

)

states = ['ANDAMAN & NICOBAR ISLANDS','ARUNACHAL PRADESH','ASSAM & MEGHALAYA','NAGA MANI MIZO TRIPURA','SUB HIMALAYAN WEST BENGAL & SIKKIM','GANGETIC WEST BENGAL','ORISSA','JHARKHAND','BIHAR',

'WEST UTTAR PRADESH','UTTARAKHAND','HARYANA DELHI & CHANDIGARH','PUNJAB','HIMACHAL PRADESH','JAMMU & KASHMIR','WEST RAJASTHAN','WEST MADHYA PRADESH','GUJARAT REGION',

'KONKAN & GOA','MADHYA MAHARASHTRA','CHHATTISGARH','COASTAL ANDHRA PRADESH','TELANGANA', 'TAMIL NADU', 'COASTAL KARNATAKA', 'KERALA', 'LAKSHADWEEP']

JSflood\_pipeline = (

idf[

(idf.YEAR <= year\_slider) &

(idf.SUBDIVISION.isin(states))

]

.groupby(['SUBDIVISION', 'YEAR'])[yaxis\_JunSep].mean()

.to\_frame()

.reset\_index()

.sort\_values(by='YEAR')

.reset\_index(drop=True)

)

JSflood\_plot = JSflood\_pipeline.hvplot(x= 'YEAR',by='SUBDIVISION', y=yaxis\_JunSep,line\_width=1, title="Flood By State for JUN TO SEP")

JSflood\_plot

yaxis\_OctDec = pn.widgets.RadioButtonGroup(

name = 'Y axis',

options=['Oct-Dec', 'ANNUAL',],

button\_type='success'

)

states = ['ANDAMAN & NICOBAR ISLANDS','ARUNACHAL PRADESH','ASSAM & MEGHALAYA','NAGA MANI MIZO TRIPURA','SUB HIMALAYAN WEST BENGAL & SIKKIM','GANGETIC WEST BENGAL','ORISSA','JHARKHAND','BIHAR',

'WEST UTTAR PRADESH','UTTARAKHAND','HARYANA DELHI & CHANDIGARH','PUNJAB','HIMACHAL PRADESH','JAMMU & KASHMIR','WEST RAJASTHAN','WEST MADHYA PRADESH','GUJARAT REGION',

'KONKAN & GOA','MADHYA MAHARASHTRA','CHHATTISGARH','COASTAL ANDHRA PRADESH','TELANGANA', 'TAMIL NADU', 'COASTAL KARNATAKA', 'KERALA', 'LAKSHADWEEP']

ODflood\_pipeline = (

idf[

(idf.YEAR <= year\_slider) &

(idf.SUBDIVISION.isin(states))

]

.groupby(['SUBDIVISION', 'YEAR'])[yaxis\_OctDec].mean()

.to\_frame()

.reset\_index()

.sort\_values(by='YEAR')

.reset\_index(drop=True)

)

ODflood\_plot = ODflood\_pipeline.hvplot(x= 'YEAR',by='SUBDIVISION', y=yaxis\_OctDec,line\_width=1, title="Flood By State for OCT TO DEC")

ODflood\_plot

JFAVG\_vs\_ANNUAL\_flood\_pipeline = (

idf[

(idf.YEAR == year\_slider) &

(~(idf.SUBDIVISION.isin(states)))

]

.groupby(['SUBDIVISION', 'YEAR','ANNUAL'])['Jan-Feb'].mean()

.to\_frame()

.reset\_index()

.sort\_values(by='YEAR')

.reset\_index(drop=True)

)

JFAVG\_vs\_ANNUAL\_scatterplot = JFAVG\_vs\_ANNUAL\_flood\_pipeline.hvplot(x='ANNUAL',y='Jan-Feb',by='SUBDIVISION',size=80, kind="scatter",alpha=0.7, legend=False, height=500, width=500)

JFAVG\_vs\_ANNUAL\_scatterplot

MMAVG\_vs\_ANNUAL\_flood\_pipeline = (

idf[

(idf.YEAR == year\_slider) &

(~(idf.SUBDIVISION.isin(states)))

]

.groupby(['SUBDIVISION', 'YEAR','ANNUAL'])['Mar-May'].mean()

.to\_frame()

.reset\_index()

.sort\_values(by='YEAR')

.reset\_index(drop=True)

)

MMAVG\_vs\_ANNUAL\_scatterplot = MMAVG\_vs\_ANNUAL\_flood\_pipeline.hvplot(x='ANNUAL',y='Mar-May',by='SUBDIVISION',size=80, kind="scatter",alpha=0.7, legend=False, height=500, width=500)

MMAVG\_vs\_ANNUAL\_scatterplot

JSAVG\_vs\_ANNUAL\_flood\_pipeline = (

idf[

(idf.YEAR == year\_slider) &

(~(idf.SUBDIVISION.isin(states)))

]

.groupby(['SUBDIVISION', 'YEAR','ANNUAL'])['Jun-Sep'].mean()

.to\_frame()

.reset\_index()

.sort\_values(by='YEAR')

.reset\_index(drop=True)

)

JSAVG\_vs\_ANNUAL\_scatterplot = JSAVG\_vs\_ANNUAL\_flood\_pipeline.hvplot(x='ANNUAL',y='Jun-Sep',by='SUBDIVISION',size=80, kind="scatter",alpha=0.7, legend=False, height=500, width=500)

MMAVG\_vs\_ANNUAL\_scatterplot

ODAVG\_vs\_ANNUAL\_flood\_pipeline = (

idf[

(idf.YEAR == year\_slider) &

(~(idf.SUBDIVISION.isin(states)))

]

.groupby(['SUBDIVISION', 'YEAR','ANNUAL'])['Oct-Dec'].mean()

.to\_frame()

.reset\_index()

.sort\_values(by='YEAR')

.reset\_index(drop=True)

)

ODAVG\_vs\_ANNUAL\_scatterplot = ODAVG\_vs\_ANNUAL\_flood\_pipeline.hvplot(x='ANNUAL',y='Oct-Dec',by='SUBDIVISION',size=80, kind="scatter",alpha=0.7, legend=False, height=500, width=500)

ODAVG\_vs\_ANNUAL\_scatterplot

yaxis\_flood\_mon = pn.widgets.RadioButtonGroup(

name='Y axis',

options=['JAN','FEB','MAR','APR','MAY','JUN','JUL','SEP','OCT','NOV','DEC'],

button\_type='success'

)

states\_excl\_INDIA = ['ANDAMAN & NICOBAR ISLANDS','ARUNACHAL PRADESH','ASSAM & MEGHALAYA','NAGA MANI MIZO TRIPURA','SUB HIMALAYAN WEST BENGAL & SIKKIM','GANGETIC WEST BENGAL','ORISSA','JHARKHAND','BIHAR',

'WEST UTTAR PRADESH','UTTARAKHAND','HARYANA DELHI & CHANDIGARH','PUNJAB','HIMACHAL PRADESH','JAMMU & KASHMIR','WEST RAJASTHAN','WEST MADHYA PRADESH','GUJARAT REGION',

'KONKAN & GOA','MADHYA MAHARASHTRA','CHHATTISGARH','COASTAL ANDHRA PRADESH','TELANGANA', 'TAMIL NADU', 'COASTAL KARNATAKA', 'KERALA', 'LAKSHADWEEP']

flood\_mon\_bar\_pipeline = (

idf[

(idf.YEAR == year\_slider) &

(idf.SUBDIVISION.isin(states\_excl\_INDIA))

]

.groupby(['YEAR', 'SUBDIVISION'])[yaxis\_flood\_mon].sum()

.to\_frame()

.reset\_index()

.sort\_values(by='YEAR')

.reset\_index(drop=True)

)

flood\_mon\_bar\_plot = flood\_mon\_bar\_pipeline.hvplot(kind='bar',

x='SUBDIVISION',

y=yaxis\_flood\_mon,

title='FLOOD \_ MON')

flood\_mon\_bar\_plot

#Layout using Template

template = pn.template.FastListTemplate(

title='Rainfall Data Dashboard',

sidebar=[pn.pane.Markdown("# Flood and Data of Rainfall"),

pn.pane.Markdown("####Floods are the most frequent type of natural disaster and occur when an overflow of water submerges land that is usually dry. Floods are often caused by heavy rainfall, rapid snowmelt or a storm surge from a tropical cyclone or tsunami in coastal areas "),

pn.pane.PNG('flood.png', sizing\_mode='scale\_both'),

pn.pane.Markdown("## Settings"),

year\_slider],

main=[pn.Row(pn.Column(yaxis\_JanFeb,JFflood\_plot.panel(width=700), margin=(0,25))),

pn.Row(pn.Column(yaxis\_MarMay,MMflood\_plot.panel(width=700), margin=(0,25))),

pn.Row(pn.Column(yaxis\_JunSep,JSflood\_plot.panel(width=700), margin=(0,25))),

pn.Row(pn.Column(yaxis\_OctDec,ODflood\_plot.panel(width=700), margin=(0,25))),

pn.Row(pn.Column(JFAVG\_vs\_ANNUAL\_scatterplot.panel(width=600), margin=(0,25))),

pn.Row(pn.Column(MMAVG\_vs\_ANNUAL\_scatterplot.panel(width=600), margin=(0,25))),

pn.Row(pn.Column(JSAVG\_vs\_ANNUAL\_scatterplot.panel(width=600), margin=(0,25))),

pn.Row(pn.Column(ODAVG\_vs\_ANNUAL\_scatterplot.panel(width=600), margin=(0,25))),

pn.Row(pn.Column(yaxis\_flood\_mon, flood\_mon\_bar\_plot.panel(width=600)))],

accent\_base\_color="#88d8b0",

header\_background="#88d8b0",

)

# template.show()

template.servable();

Graphical user interface

Description automatically generatedGraphical user interface, chart

Description automatically generatedGraphical user interface, application

Description automatically generated