

TEAM DELIVERABLE 3 DOCX- GROUP11 SE PROJECT

Functional Requirements:

- 1) The whole Prediction page thing which user will see is on the web with designated localhost.
- 2) When the user clicks the link, it prompts to the prediction page screen.
- 3) The prediction page screen is created using HTML and CSS
- 4) The user interface will be a web page that user can see the prediction page.
- 5) The web implementation is flask web framework which was developed with python code
- 6) On the flask-based implementation web there will be options to select the state for which state you are predicting the flood.
- 7) There user can input the values for respective months to predict the values.
- 8) The flood prediction will be done by clicking the Predict button.
- 9) When the user clicks the Predict button it will show flood happen or flood does not happen based on the inputted values.

PHASES:

We have divided our project work into three phases as the following

PHASE1:

In phase 1 we pre-process the data with the validation techniques. Several data cleaning tasks will be done using Panda's library in python. The missing values or outliers will be found by these data cleaning process and visualizing the collected data.

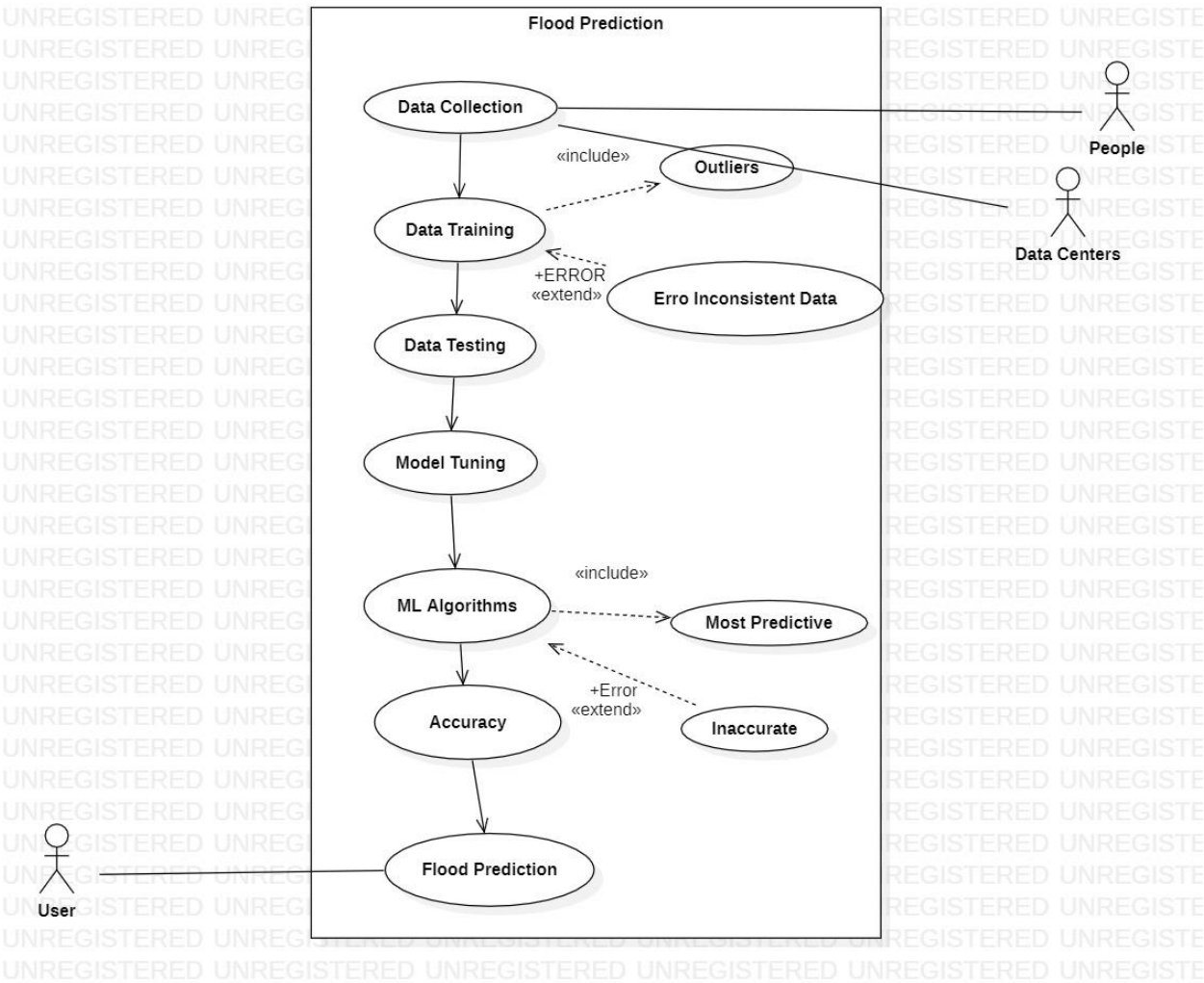
PHASE2:

In phase 2 we do the performance measurements for the ML algorithms Logistic Regression, Decision Tree, Random Forest algorithm.

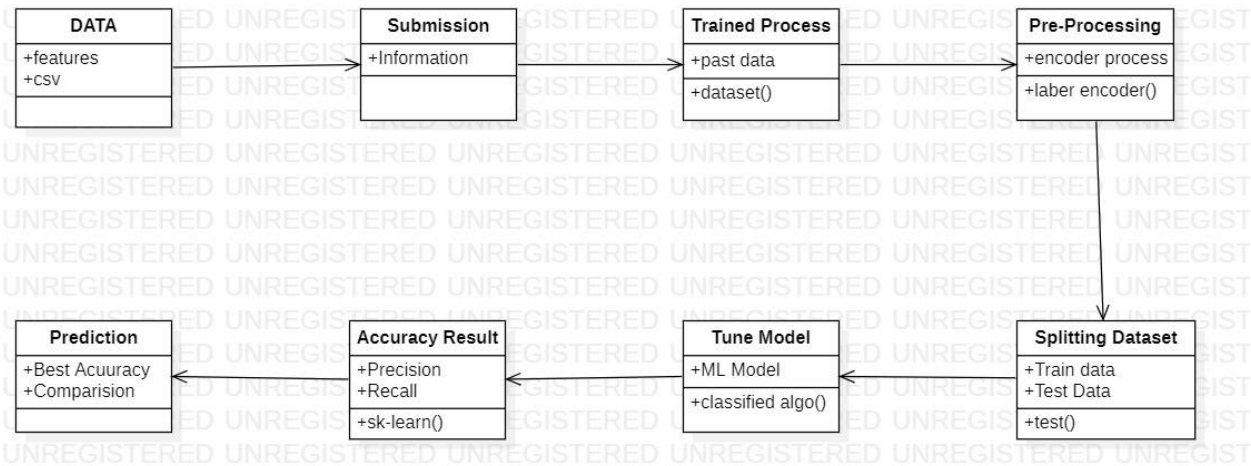
PHASE3:

In phase 3 we implement our code using flask web framework which is written python. Using the local host address the source code is executed and flood prediction will be done.

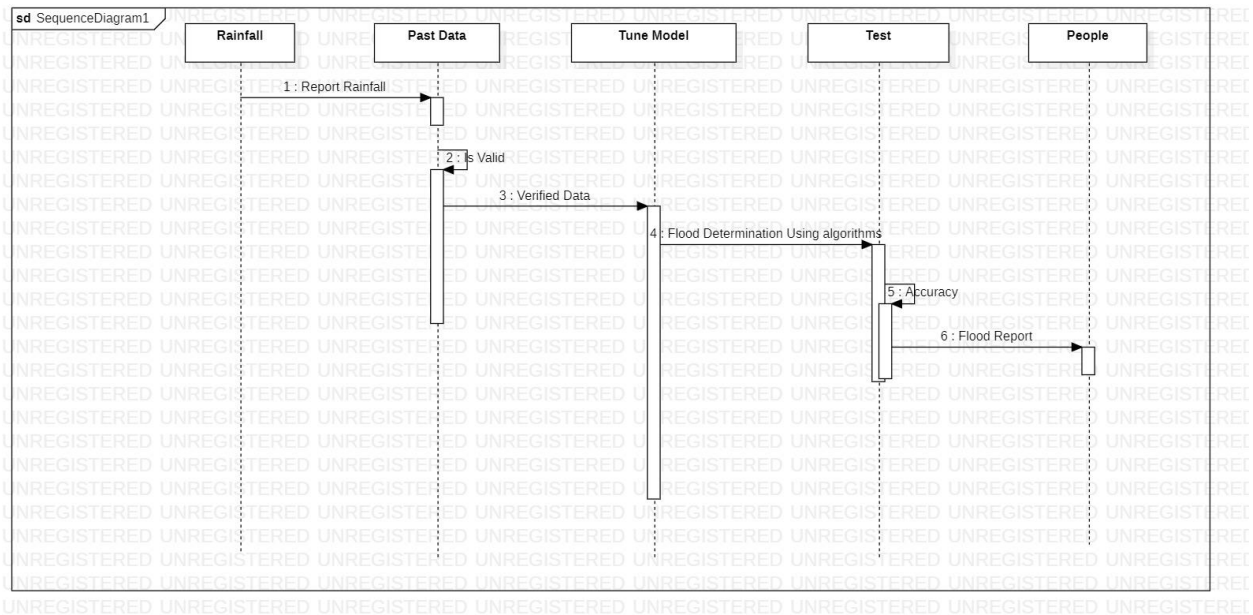
USE CASE Diagram:



CLASS Diagram:



SEQUENCE Diagram:



Report Test Cases from the User End:

Test Case Type	Description	Test step	Expected result	Status of Test case
Functionality	The Provided html page is working with CSS and source code	Execute the html first and next with CSS and source	The Prediction page should be viewed	Pass
	Is the flask app working properly	Executing the deploy flask related code	Prediction Page on web	Pass
Security	Copying designated address of local host and trying it in another system	Copy the flask app provided address	The page should not be accessed like that	Pass
	Using different Port numbers	Give default port number	It should execute on that port number	Pass
	Executing simultaneous files	Run same file simultaneously	It does not execute like that	Pass
Usability	Ensure the links work properly for User	Click the link	Prompt to web	Pass

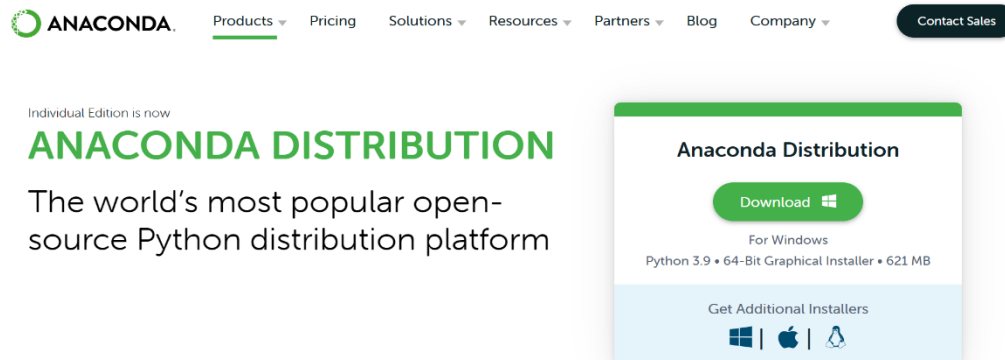
USER MANUAL:

Installation:

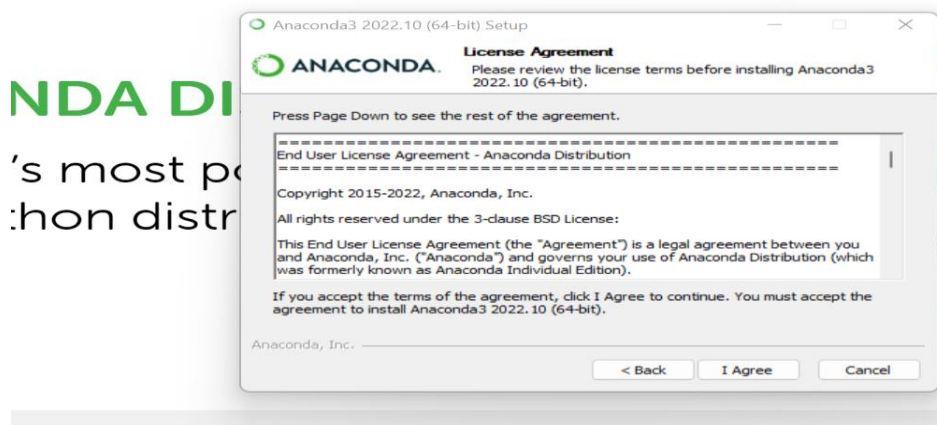
With the help of the desktop graphical user interface (GUI) Anaconda Navigator, which is a part of the Anaconda® distribution, you can simply manage conda packages, environments, and channels as well as run applications. Navigators have access to local Anaconda Repositories as well as Anaconda.org to search for packages.

Steps to install Anaconda Navigator.

- Download Anaconda wha



- Double click on the installer when downloaded.
- Click next and agree the license



- Select Just for me unless you want to install for all users.
- Select the destination folder for anaconda

- Select the path environment variable. Adding anaconda to your path environment variable is not recommended, because it may interfere with other software.
- Choose anaconda as default browser.

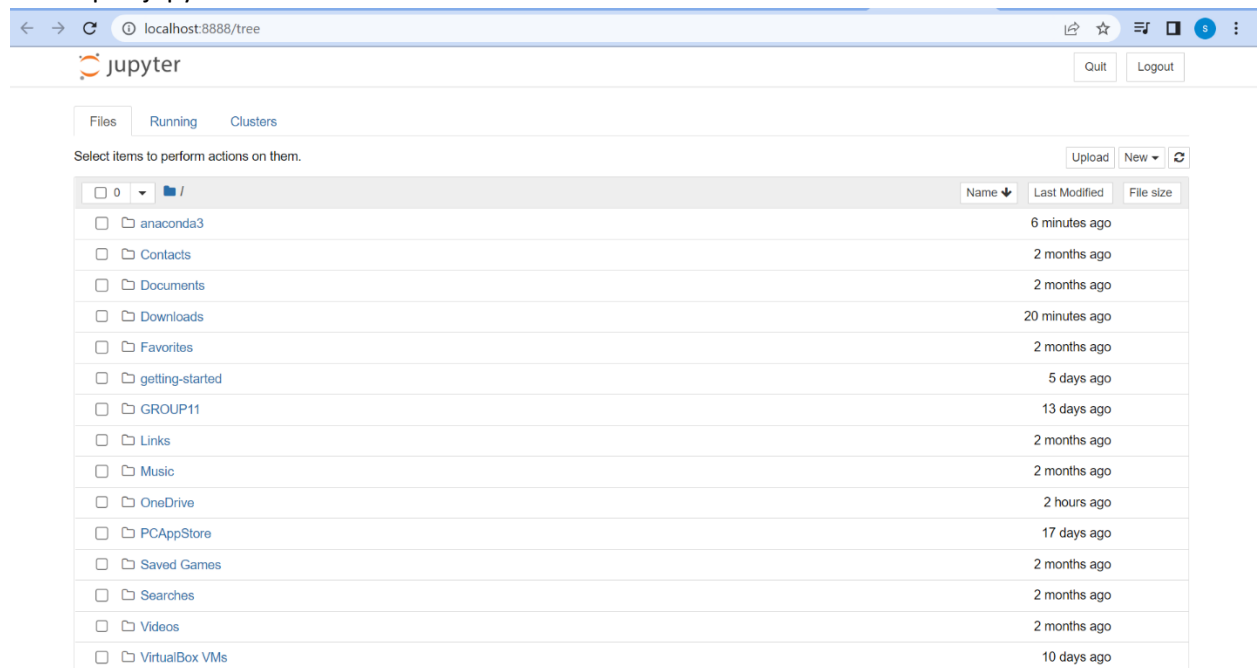
- Install anaconda and open anaconda prompt

Type jupyter notebook and run, so that the jupyter can be launched.

```
(base) C:\Users\katam>jupyter notebook
[I 18:33:55.890 NotebookApp] Writing notebook server cookie secret to C:\Users\katam\AppData\Roaming\jupyter\runtime\notebook_cookie_secret
[I 2022-10-23 18:34:03.147 LabApp] JupyterLab extension loaded from C:\Users\katam\anaconda3\lib\site-packages\jupyterlab
[I 2022-10-23 18:34:03.147 LabApp] JupyterLab application directory is C:\Users\katam\anaconda3\share\jupyter\lab
[I 18:34:03.156 NotebookApp] Serving notebooks from local directory: C:\Users\katam
[I 18:34:03.156 NotebookApp] Jupyter Notebook 6.4.12 is running at:
[I 18:34:03.156 NotebookApp] http://localhost:8888/?token=19fef899c23d4fcd424a7c90f2a8c831112b0ad72f35ba71
[I 18:34:03.156 NotebookApp] or http://127.0.0.1:8888/?token=19fef899c23d4fcd424a7c90f2a8c831112b0ad72f35ba71
[I 18:34:03.156 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 18:34:03.273 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/katam/AppData/Roaming/jupyter/runtime/nbserver-11180-open.html
Or copy and paste one of these URLs:
http://localhost:8888/?token=19fef899c23d4fcd424a7c90f2a8c831112b0ad72f35ba71
or http://127.0.0.1:8888/?token=19fef899c23d4fcd424a7c90f2a8c831112b0ad72f35ba71
```

- Then open jupyter notebook



- Installation of anaconda is done.

Running the jupyter notebook:

The Jupyter Notebook App can be opened by clicking on the Anaconda-installed Jupyter Notebook icon in the start menu (Windows) or by entering the following commands in a terminal (cmd on Windows): Python Notebook

This will create a new browser window (or new tab) with the Notebook Dashboard, a kind of control panel that lets you choose which notebook to open among other things. The Jupyter Notebook App can only access files in its start-up folder when it is launched (including any sub-folder). If you save your notebooks in your home folder or any subfolders, no setting is required. Otherwise, you must select a start-up folder for the Jupyter Notebook App that will house all of the notebooks.

Save notebooks:

The notebooks automatically save changes every few minutes. Make a copy of the notebook document (menu file -> make a copy...) and save the edits there to prevent making changes to the original notebook.

Executing a notebook:

- Place the notebook you want to use in your notebook folder after downloading it (or a sub-folder of it).
- Open the Jupyter Notebook application.
- Navigate to the notebook in the dashboard of the Notebook The tab will be opened when you click it.
- For a description of the notebook use the menu option Help -> User Interface Tour. It will help you to know about the Jupyter Notebook.
- To run one cell at a time just hold shift and enter it can run step by step.
- By selecting the Cell -> Execute All option from the menu, you can run the entire notebook in one go.
- Click on the menu item Kernel -> Restart for restarting the kernel which is also known as the computational engine. This can be helpful if you want to rerun a calculation from scratch (by deleting variables or closing open files, for example).

Compilation Instructions

Anaconda navigator is required to compile the Python code that we have chosen to write.

Step1: Launch Anaconda Navigator

Step2: launch Jupyter Notebook

Step3: In Jupyter notebook open a new python file

Step4: Copy the each individual developed code into respective modules

Step5: Upload the data set into jupyter notebook

// while running the ipynb files in different modules following working takes place

- **Module 1:** Data preprocessing / Cleaning
- Data cleaning tasks are performed using Pandas library in python, with a particular emphasis on cleaning tasks and outliers. It prefers less time in cleaning data and spend more time in experimenting and modeling.
- Finally, this converts raw data into clear and understandable data
- **Module 2:** Data Visualization
- The entire data set that we cleaned and filtered is represented graphically for easy comprehension of the flow.
- **Module 3:** Logistic regression
- The data that we collected will be processed and displayed in the form of estimated models during this process. In general, this displays binary outcomes such as yes or no based on previous data sets collected.
- **Module 4:** Decision-tree
- This reads the data and divides it into parts; based on the results, this makes decisions and implements the algorithm. The data is split until it reaches the termination condition.
- **Module 5:** Random forests classifier
- The classification and regression will be done it constructs the trees from the various samples of data.

Step6: Deployment: Flask Framework

After successfully running the modules, when we run the deployment file, we will be presented with a link to the file's location, i.e., local host link.

Step7: After clicking the link, we will be shown the state name, months, and annual blocks. Then we must select the state, month, and other details. After that, we should then click on predict, and the user will be notified based on the input data.

Step8: Finally, the flood will be predicted using machine learning algorithm.

Report Review:

Asked to update the functional requirements, asked to mention the phases of the project.

Report Reflection:

The advised changes have been added to the project. The functional requirements and using flask web framework have been added to the project.

Description on what we accomplished, what we could improve:

This part of the project requires dataset which is collection of different areas atmosphere conditions. Here the temperatures are taken as input, and it predicts whether the flood will happen or not. In this

case we are going to weather conditions, but we can even use pictures of the climate and use different algorithm as the same way we are using now to predict the result.

Contribution Table:

Member	Contribution	Description
Srikanth Pavuluri	14	Report, code
Venkat Subbareddy Kattamedi	14	Report, code
Chandana shivannagari	12	Report, code
Priya Kuppireddy	12	Report, code
Madhuri sri yarramreddy	12	Report, code
Kalyan Kumar Goparaju	12	Report, code
Mamatha Amireddy	12	Report, code
Mahesh Reddy	12	Report, code