**TEAM DELIVERABLE 4 DOCX- GROUP11 SE PROJECT**

**Functional Requirements:**

|  |  |  |  |
| --- | --- | --- | --- |
| # | FR name | FR description | phase |
| FR1 | Data Selection | The **user** can **choose** the current or **present data** | Phase 1 |
| FR2 | Prediction history | The system shall render the 3 recent predictions made by the user. | Phase 1 |
| FR3 | Dashboard | The system shall display … | Phase 2 |
| FR4 | Share prediction | The user can share the prediction via email | Phase 2 |
| FR5 | Attribute Selection | The **user** can **select** the **state** and **month** of the **prediction**. | Phase 3 |
| FR6 | Data Entry | The user can enter the recorded data of rainfall. | Phase 3 |
| FR7 | User clicks | The user can predict by entering values and clicking the predict interface | Phase 3 |

**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: (Description)**

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

FR3:

The dashboard contains of several components that functionality be like selecting the states from the provided list of states, set of months are provided to input the rainfall occurred, an annual record data input, a click button of prediction button that user clicks on to give the output for inputted data by the user. The pop up will occur with text of “Flood Happen, Flood Not Happen”.

FR4:

The User can share the prediction result via mail or through link share. The results can be viewed by others to know about the prediction.

In phase 3 we have done the performance measurements of the algorithms we have used libraries to visualize the performance given by the algorithms. The data was tested with the algorithms to know the accuracy.

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.

**CLASS Diagram:**

**Graphical user interface, application, table

Description automatically generated**

**SQUENCE Diagram:**

**A picture containing diagram

Description automatically generated**

**Use Case Diagram:**

**Diagram

Description automatically generated**

**Report Test Cases from the User End:**

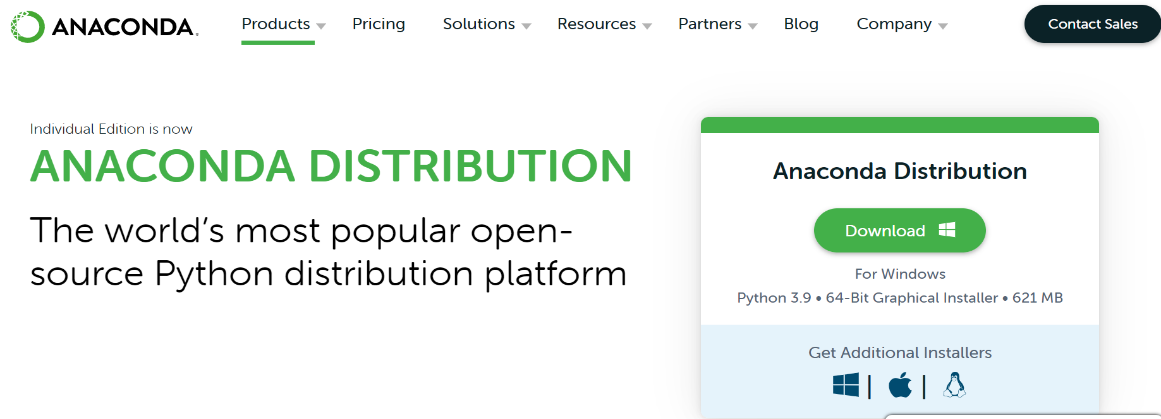
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Case Type | Description | Test step | Expected result | Status of Test case |
| Functionality | The Provided html page is working with CSS and source code | Execute the html first and next with CSS and source | The Prediction page should be viewed | 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 |
|  | Enter blank values | Leave blank values for user input fields. | Ask to must fill the fields | 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 licenseGraphical user interface, text, application, chat or text message, email

  Description automatically generated
* 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.

Graphical user interface

Description automatically generated

* Then open jupyter notebook Graphical user interface, text, application, email

  Description automatically generated
* 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 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.

* **Module 6:** support Vector Machine

The line will be plotted using the data divided into parts based on the results the prediction will be done

**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 Code Inspection:**

**Feedback meeting**: 25 minutes

**Team meeting summary:**

* We discussed and read the code of our pair group and pointed out some things in their file.
* We decided to tell them to improve the code convention.
* To minimize the warnings in the code.

**Feedback meeting summary:**

* They suggested us to add one more algorithm
* And test the data with that algorithm.
* They suggested to improve the code convolution.
* One of our teammates asked about their modules counterrally they asked the same question.
* Answered the questions raised by them and us.
* Asked about how many modules are still needed to implement.

**Things accepted:**

* Decided to train one more algorithm with the collected data.
* And improve the code convention.

**Report Reflection:**

The advised changes have been added to the project. The Support vector machine algorithm is used to test the data and get an accuracy result.

**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 past data is taken as input, and it predicts whether the flood will happen or not. We have improved our deliverable 4 with feedback of our teaching assistant.

**Contribution Table:**

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