**System Structure :**

MODEL

Data Processing

Classification of ML Algorithms

Trained dataset

Flood Dataset

Test

Dataset

The System Structure consists of these main processes collecting the flood dataset, data preprocessing, test dataset, Trained data, classification of ML Algorithms, Preparing model.

* **Flood Dataset:**

The data has been collected from sources and the data sources will be a metrological department provided data, Data source from the internet, social media data regarding flood events which were find out from the Kaggle website.

* **Data Processing:**

The data may consist of lot of errors like outliers, missing data values, the data may input wrongly, inconsistent data, loss of data while transferring, forgot the data to enter. The libraries were loaded with the dataset. With data shape data type the variable identification will be done. The noisy data is cleared, the removal of errors will result a best value for making decision. The data will be visualized as graphs and plots

* **Training Dataset:**

As we know that the training data will be used by machine learning algorithms to learn from the obtained data in datasets. From these data the ML algorithms will find patterns, they make decisions and evaluate those decisions. The training data is a large set data compared to test dataset this is so because of when we feed more data for training then the model learns the mot and make decisions appropriate.

* **Test Dataset:**

Test Dataset is a never seen dataset which is used to test the model. The data is called testing data this can be used for performance evaluation, algorithm’s progress, and it’s training. It must represent the actual dataset and when the dataset is large then the meaningful predictions will be generated. It just confirms whether it works or not with unseen data.

* **Classification Of ML algorithms:**

The supervised Machine learning algorithms were mostly used for classification and regression problems. The classification problems like speech recognition, handwriting recognition, the verification things like biometric. In supervised machine learning the models will learn from the past data or labelled data. It decides how the data should be shared based on the patterns.

1. Logistic Regression:

The dataset is analyzed by the algorithm where the outcome will be decided by the one or more independent variables the relation between the dependent and independent variables is known that is the main goal of logistic regression algorithm. The prediction of the categorical dependent variable is done and the data in it was coded with 1 and 0 to represent the true and false respectively

1. Decision Tree:

Decision tree is one of the most popular and powerful algorithms. The algorithm can handle both he continuous and categorical output variables. The decision tree algorithm will assume the whole training set as root and the attributes are assumed as categorical and continuous output variables and those were distributed based on attribute values recursively. The algorithm builds the models in the structure of a tree. A decision node consists of two or more branches and a leaf node represents decision or classification.

1. Random Forest Classifier:

The algorithm will pick n number of data records from the given dataset and builds a decision tree for the randomly choose data records. A value is predicted from each tree in the forest for the new record and the value is output and the average of these values will be taken as average of values for trees predicted.

**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.

**2.Project Requirements:**

**General:**

Requirements are the basic constraints which were collected while designing the system and these are required to develop the system. These following discussed are the requirements.

1. Functional Requirements
2. Non-Functional Requirements
3. Environmental Requirements

* **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.

**NUMPY:**

NumPy provides a high-performance multidimensional array object, and tools for working with these arrays. It links to a variety of other packages for doing advanced calculations in science and engineering, including pulse and spatial processing, integration, linear algebra, statistics, random number generation, Fourier transforms and image processing.

**Sk-Learn:**

Sk-learn is a Python module that provides a collection of machine learning tools. It features various classification, regression and clustering algorithms including support vector machines, boosting, k-nearest neighbors and decision trees. Sk-learn also implements active learning, cross validation, model selection and A/B testing.

**Pandas:**

Pandas is an incredible library built on two powerful Python packages: Numpy and Scipy. It provides a high-performance, easy-to-use data manipulation, conversion, and analysis library for use in data science applications that are both fully open source and free for commercial use.

**Matploit:**

The matploit library is a python library that allows you to run binary analysis on a raw file without the need for an emulator. Matploit library for Generating, Running and Evaluating Inverse Probability Weighted Models with Missing Data.

**Seaborn:**

seaborn is a Python library for making attractive and informative statistical plots. It provides a high-level interface to drawing packages like matplotlib while simultaneously giving you full control over your plot’s aesthetics. seaborn aims to make exploratory data analysis fun & easy.

* **Non-Functional Requirements:**

There are some processes in functional steps:

1. Problem finding
2. Data Preparation
3. Algorithm Evaluation
4. Algorithm Improvement
5. Result Prediction

* **Environmental Requirements:**

1. Software Requirements:

OS: Windows

Tool: Jupyter Notebook On anaconda

1. Hardware requirements:

Processor: Pentium IV/III

Hard disk: 80 GB Storage

Ram: 2 GB Minimum

**3.Development phase plan:**

The development phase plan starts from gathering the data, Loading the dataset, importing the library packages, analyzing the general properties, finding the outliers, checking the unique and count values, exploration analysis used for comparing and predicting the result.

* **Data Wrangling**

The section handles the data loading, checking the data is cleansed, the dataset will be cleaned and analyzed. It makes sure that the document is justified for cleaning

* **Data Collection**

The data collected is divided as training and testing sets. The datasets were applied to the algorithms Decision Tree algorithm, Random Forest algorithm, Logistic Regression algorithm for the prediction and for the test result accuracy.

* **Preprocessing**

When the data was collected it has to be preprocessed because it may consist of outliers, missing values the data will be preprocessed for better results the outliers will be removed

* **Constructing a Predictive Model:**

The data which is training dataset and test dataset will be used and tested by each model and the best accuracy produced will be chosen and using the algorithm the prediction will be given.

Data Gathering

Data Pre Processing

Choosing Model

Train Model

Tune Model

Test Model

Prediction

kjkk

* **List Of Modules Implementing**

1. Pre- Processing the data

2. Visualizing the data

3. Algorithm comparison for predicting the best accuracy result

4. Flask deployment

* **Software Description**

Anaconda Navigator is a free and open-source distribution of the python and R languages. There are lot of applications in the anaconda navigator by default. In that we will be using the Jupyter Notebook we can install it using commands in the shell and we can launch through the same by just simply clicking the icon through desktop or in the file location or it can be launched through terminal or command line. The file extension will be in ipynb. There are several steps when you are applying machine learning models to the data sets, they are:

* Problem definition
* Data Preparation
* Algorithm evaluation
* Result Improvement
* Result presentation

The data loading , briefing the data and evaluating the algorithms and make decisions these are main things in the machine learning project.

1. Python Installation
2. Dataset Loading
3. Dataset briefing
4. Dataset Visualization
5. Algorithm evaluation
6. Making Prediction

* **Deployment**

The deployment will be using flask (web flask framework). Flask is a micro web framework written in python. It does not require any files. It was designed to be easy to use and extend. It gives so much more control on the development.

**Advantages using Flask:**

1.Higher compatibility with latest technologies.

2.Technical experimentation.

3.Easier to use for simple cases.

4.Codebase size is relatively smaller.

5.High scalability for simple applications.

6.Easy to build a quick prototype.

7.Routing URL is easy.

8.Easy to develop and maintain applications.

|  |  |  |  |
| --- | --- | --- | --- |
| Member name | Contribution description | Overall Contribution | NOTE |
| Srikanth P | Report and Documenting | 12.5 | System Structure |
| Venkat K | Report and GitHub | 12.5 | Datasets |
| Kalyan K | Report | 12.5 | Classification of algorithms |
| Mahesh | GitHub and Modules implementing | 12.5 | Adding meeting minutes Functional requirements |
| Mamatha | Report | 12.5 | Non-Functional requirements |
| Priya | Report | 12.5 | Software description |
| Chandana | Report | 12.5 | Environmental requirements |
| Madhuri Sri | Report | 12.5 | Software Description |