**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 validation techniques will be used to get the error rate of the ML models used. Data collection, data analysis and the process of finding the data content and quality and structure can add up to a time-consuming to-do list. There may be a lot of errors like the type of missing values these may be occurred because of user may be forgot to fill out the data, data, a lost while transferring manually from any od=f the database or may be because of the programming errors. The libraries packages were loaded with the given dataset. The analysis will be done by variable identification by data shape, type of data, evaluating the missing values. The steps of cleaning or processing the dataset may vary from another dataset but here the primary goal was to remove errors and irregularities to increase the data value in decision making. The dataset will be a removed noisy data which is used for the visualizing the data obtained after processing the data. The data may not be that much worthy, or it does not make any sense if it is not represented in a visual form like charts, graphs, 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:**

Here the Supervised Machine Learning algorithms were used for classification and regression problems. The examples of some classification problems are speech recognition, handwriting recognition, biometric, document verification etc. In Supervised learning the models will learn from the labelled data. After knowing the data, the algorithm decides to which label the data should be given and that will be based on the patterns.

1. Logistic Regression:

The algorithm analyzes a dataset in which there where one or more independent variables that determines an outcome. The outcome is measured with a dichotomous variable. The main goal for the logistic regression is to find out and describe the relation between a set of independent variables and dependent variables. The algorithm is used to predict the probability of categorical dependent variable. In logistic regression algorithm the dependent variable is a binary variable and the data contained in it was coded with “1” which represents true or success or “0” which represents false or failure.

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 random forest algorithm will randomly pick the n number of records from the provided data set and build a decision tree based on these randomly picked n data records. The number trees will be chosen, and the process is repeated and finally for a new record each tree in the forest predicts a value for Y as output and the output value can be calculated by taking the average of all values which the trees predicted.

**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:**

The technical requirements are the software requirements for the software product. It is the first step in the requirement analysis process. It consists of the requirement of software system. The given details follow the special libraries such as sk-learn, pandas, NumPy, matplotlib and seaborn.

**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 defines
2. Preparing data
3. Evaluating algorithms
4. Improving algorithms
5. Prediction the result

* **Environmental Requirements:**

1. Software Requirements:

Operating System: Windows

Tool: Anaconda with Jupyter Notebook

1. Hardware requirements:

Processor: Pentium IV/III

Hard disk: minimum 80 GB

RAM: minimum 2 Gb

**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, Uni-variate analysis and exploration data analysis of bi variate and multi-variate, comparing the algorithm for predicting the result.

* **Data Wrangling**

In this section of the report will load in the data, check for cleanliness, and then trim and clean given dataset for analysis. Make sure that the document steps carefully and justify for cleaning decisions.

* **Data Collection**

The data set collected for predicting given data is split into Training set and

Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data Model which was created using Random Forest, logistic, Decision tree algorithms and Support vector classifier (SVC) are applied on the Training set and based on the test result accuracy, Test set prediction is done.

* **Preprocessing**

The data which was collected might contain missing values that may lead to inconsistency. To gain better results data need to be preprocessed to improve the efficiency of the algorithm. The outliers must be removed, and variable conversion need to be done.

* **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.Data Pre-processing

2.Data Analysis of Visualization

3.Comparing Algorithm with prediction in the form of best accuracy result

4.Deployment Using Flask

* **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. When you are applying machine learning to your own datasets, you are working on a project. A machine learning project may not be linear, but it has several well-known steps:

* Define Problem.
* Prepare Data.
* Evaluate Algorithms.
* Improve Results.
* Present Results.

The best way to really come to terms with a new platform or tool is to work through a machine learning project end-to-end and cover the key steps. Namely, from loading data, summarizing data, evaluating algorithms and making some predictions.

Here is an overview of what we are going to cover:

1. Installing the Python anaconda platform.
2. Loading the dataset.
3. Summarizing the dataset.
4. Visualizing the dataset.
5. Evaluating some algorithms.
6. Making some predictions.

* **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 |