**1.INTRODUCTION**

The biggest issue for flood risk management in urban areas is the prediction that under climate change there will be considerably more flooding in these areas. Specifically, in the UK, flooding is considered to be one of the biggest problems that the country is facing today, with climate projections suggesting that the increase in total rainfall will provoke major, more frequent and less predictable flood events. The impact of floods on housing is also increasing due to the ongoing development of settlements in flood-prone areas, together with the rising vulnerability of assets to risk . The phenomenon of flooding is extremely complex and subject to change. Incidents are no longer restricted to obvious areas where a river or stream exists; many urban floods are simply caused by huge amounts of rain falling very quickly (flash floods) in an area where the drainage system is unable to cope or due to unexpected underground basin recharge and rise of the groundwater levels. As a consequence, there is an emerging motivation to understand how accurate our knowledge can be about flood risk—its location, timing and duration—and how data collection and analysis can assist us. People usually make an effort to stay aware of what happens in their neighborhoods, especially if their health, wellbeing or prosperity is at stake. It has already been shown that public information sufficiency, risk perception and self-efficacy can predict risk-information seeking behaviour. As a particular type of information seeking, Web search has become a key reference source. Although not always 100% accurate, it is fast and free of charge. Built-in traffic monitoring engines within websites help to collect data about information-seeking behaviour and one such example is Google Analytics, which is capable of tracking hourly information about unique page visits, number and duration of individual sessions at the level of cities and towns and—in some cases—at the level of postcodes. Google Analytics has found wide applications in business and website optimization , and is increasingly becoming one of the main generators of ‘big data’ records, alongside other systems that record, for example, our communications, travel and retail activities. A number of recent studies have provided evidence that ‘big data’ can reveal a great deal about people’s real-world, collective decision-making and responses to events and can even help to predict such phenomena, e.g. Hurricane Sandy .

**1.1.MOTIVATION**

Urban flooding is a threat to a great number of cities worldwide, especially given its increasing frequency of occurrence in recent years. Its impact, including loss of life and damages to both public and private properties, can be further deepened by climate change and accelerated urbanization. In July 2019, heavy rain caused 18 deaths and triggered massive transport disruptions in Mumbai, India. Its impact, including loss of life and damages to both public and private properties, can be further deepened by climate change and accelerated urbanization (Falconer et al., 2009). This type of flooding usually occurs when intense rainfall exceeds the capacity of an urban drainage system. Recent extreme precipitation events have raised awareness from both authorities and citizens to the challenges of predicting and managing urban pluvial floods. A survey, conducted between 2008 and 2010, showed that 218 Chinese cities suffered severe urban flooding at least once, and more than 100 cities experienced it more than three times. In order to prevent flooding and its consequences, city authorities (e.g. meteorological offices, emergency agency offices or water authorities) usually need to make predictions of floods. In the UK, about 40% of damages and associated economic losses in cities are estimated to result from pluvial flooding.

**1.2.PROBLEM DEFINITION**

In order to prevent urban flooding and its consequences, city authorities (e.g. meteorological offices, emergency agency offices or water authorities) usually need to make predictions of pluvial floods. This is based on good prediction of precipitation characteristics, such as peak intensity, arrival time and duration. Many studies investigate the prediction of urban flooding by hydraulic models (Li, 2020; Li and Willems, 2019), by simulating the inundated area and depth given certain historical or predicted rainfall scenarios. Here we build a model using machine learning techniques in order to predict the floods and alert the people well in advance. It takes rainfall threshold as the input and generates the output as whether the flood occurs or not. Based on the rainfall conditions of a particular area from January to December it predicts the floods well in advance which is very helpful for the people. The rainfall dataset that is taken in this model contains rainfall from 1901-2015 and the unit of rainfall is in mm. Finally it predicts the flood occurrence before the flood occurs.

**1.3 OBJECTIVE OF THE PROJECT**

We aim to build classification model that is Random Forest Classifier and also fine tune the parameters of the model. By using this classification model, we try to predict the flood in major cities.By predicting the floods well in advance people and government can take proper measures to save their properties and lives. Our main objective is to protect the peoples lives from the dangerous floods. So, we collect the rainfall from an area and pre-process it making ready to build a model. Later the feature selection is performed and the data is fed to the random forest classifier and finally prediction is done based on the previous data available in the dataset. By this model urban flooding in major cities can be known well in advance and the various disaster management departments and people can be prepared to face the floods. By predicting the floods well in advance their properties and precious lives will be saved and they can live happily without fear and can take proper measures in order to protect themselves from floods.

**2.LITERATURE SURVEY**

# 2.1. Quantifying the flood resilience properties of walls in typical UK dwellings

**Authors: M. Escarameia, Eur Ing MSc MICE, A. Karanxha, MSc, A. Tagg, MSc MICE MCIWEM**

**Abstract:** Building for resilience against floodwater has become increasingly important given the high demand for new houses in the UK and the need to build in flood-prone areas. There are current recommendations on the use of building materials that will minimize the impact of floods; however, these are rarely substantiated by scientific evidence and tend to relate to individual building materials rather than composite constructions. To fill this gap, a laboratory investigation was undertaken aimed at quantifying the properties of walls (and their constituents) when exposed to floodwater. Time-varying leakage rates and drying rates were determined for 13 different wall constructions subjected to 1 m water depth. Both masonry and timber frame walls were studied, incorporating common types of cavity insulation. Recommendations are given for resilient construction. Practical application: The work described in this paper was part of a larger study commissioned by the Department for Communities & Local Government (DCLG) and EA/ DEFRA, aimed at obtaining a better understanding of how buildings behave when subjected to floodwater. The ultimate aim was to develop guidance for the construction of new buildings, focusing on domestic properties in areas at risk of flooding. This will be taken into account in the next revision of the Building Regulations, namely in Approved Document C. A guidance document on resilient and resistant construction, synthesising the findings of the overall project, will be a reference document for the new Planning Policy Statement (PPS25), which defines the requirements for resilient developments in England and Wales.

# 2.2Assessing the effects of flood resilience technologies on building scale

**Authors: Reinhard Schinke &Thomas Naumann**

**Abstract:** The impacts of floods on housing are increasing due to more frequent and severe weather events as well as the ongoing development of settlements in flood-prone areas together with the rising vulnerability of assets at risk. Therefore, the improvement of the resilience properties of buildings to better cope with flooding has become a key issue towards more flood resilient cities in European research in recent years. The implementation of flood resilience technologies (FReT) on the individual property scale provides a previously untapped potential to reduce flood damage to buildings due to insufficient transparency of their effects. To overcome this obstacle, the paper presents a four-step extension of a synthetic approach for flood vulnerability analysis to provide evidence on the potential effects of FReT uptake concerning flood damage mitigation. The proposed approach has been integrated in the GIS-based flood damage simulation model HOWAD to support the assessment of FReT alternatives. The simulation of flood damage to buildings in the case study Heywood, Greater Manchester (United Kingdom) revealed the potential of the extended approach to determine the consequences of FReT implementation on building scale.

# 2.3 Communicating uncertainty via probabilities: the case of weather forecasts

**Authors: Beth Proudley**

**Abstract:** Capturing uncertainty through numerical probabilistic statements is orthodoxy in risk science—and most of science and technology. There are a wide range of views on the utility of such statements for risk communication, and they are often seen as being central to the failure to generate common understanding about risks between science and non-scientists. The extent to which probability statements are understood is unclear. If such statements are misunderstood by many, what alternatives might communicate uncertainty better? These questions are examined in the context of daily weather forecasts. The probabilities used in such statements concern daily events experienced by everyone, unlike the extremely small probabilities about unfamiliar events often used in risk communication. If people do not understand weather forecasts, there is little hope that statements about unfamiliar events using unfamiliar language will be understood. Some jurisdictions use numerical probabilistic statements on the likelihood of precipitation, and a variety of qualitative or categorical forecasts are also used. Drawing on a range of sources including public surveys conducted by the Australian Bureau of Meteorology, the paper examines public understanding of probabilities and public and specialist understanding of verbal categorical forecast terms. The majority of those surveyed have basic understanding about probabilities as used in weather forecasts, but significant groups do not. However, there was limited agreement among forecasters on what the probabilistic statements meant. Similarly, there was limited shared meaning between forecasters and the public on the verbal forecast expression examined.

# 2.4Information Needs and Risk Perception as Predictors of Risk Information Seeking

# 

# Authors: Jan M. Gutteling

**Abstract:** This paper introduces a theoretical framework that describes the importance of public's information sufficiency, risk perception, and self-efficacy as predictors of intended risk information seeking behaviour. Based on theoretical assumptions, measurement instruments for relevant concepts were developed and validated using data from a mail questionnaire. Relationships among selected determinants of risk information seeking behaviour were analysed. Results indicate that information needs, risk perception, and current knowledge are direct predictors of intentions to seek information. Trust, engagement, social influence, and self-efficacy affect risk perception and the need for information is influenced by engagement and social influence.

**3. ANALYSIS**

**3.1 EXISTING SYSTEM:**

Urban flooding is significantly different from rural flooding as urbanization leads to developed catchments, which increases the flood peaks from 1.8 to 8 times and flood volumes by up to 6 times. Consequently, flooding occurs very quickly due to faster flow times (in a matter of minutes). Urban areas are densely populated and people living in vulnerable areas suffer due to flooding, sometimes resulting in loss of life. It is not only the event of flooding but the secondary effect of exposure to infection also has its toll in terms of human suffering, loss of livelihood and, in extreme cases, loss of life. Increasing trend of urban flooding is a universal phenomenon and poses a great challenge to urban planners the world over. The existing system uses physically based models to predict hydrological events, such as storm, rainfall, shallow water condition , including the coupled effects of atmosphere, ocean. However, physical models need a large volume of data and computational resources. As a result, the output of a physical model is usually case-specific. In other words, we have to run the model to make predictions for flooding during each separate rainfall scenario. The development of physically based models often requires in-depth knowledge and expertise regarding hydrological parameters, reported to be highly challenging.

Physically based models were long used to predict hydrological events, such as storm, rainfall/runoff, shallow water condition, hydraulic models of flow, and further global circulation phenomena, including the coupled effects of atmosphere, ocean, and floods. Although physical models showed great capabilities for predicting a diverse range of flooding scenarios, they often require various types of hydro-geomorphological monitoring datasets, requiring intensive computation, which prohibits short-term prediction. Furthermore, as stated in Reference, the development of physically based models often requires in-depth knowledge and expertise regarding hydrological parameters, reported to be highly challenging. Moreover, numerous studies suggest that there is a gap in short-term prediction capability of physical models. For instance, on many occasions, such models failed to predict properly. Similarly, numerical prediction models were reported in the advancement of deterministic calculations, and were not reliable due to systematic errors. Nevertheless, major improvements in physically based models of flood were recently reported through the hybridization of models, as well as advanced flow simulations. In addition to numerical and physical models, data-driven models also have a long tradition in flood modeling, which recently gained more popularity. Data-driven methods of prediction assimilate the measured climate indices and hydro-meteorological parameters to provide better insight. Among them, statistical models of autoregressive moving average (ARMA), multiple linear regression (MLR), and autoregressive integrated moving average (ARIMA) are the most common flood frequency analysis (FFA) methods for modeling flood prediction. FFA was among the early statistical methods for predicting floods. Regional flood frequency analyses (RFFA) , more advanced versions, were reported to be more efficient when compared to physical models considering computation cost and generalization. Assuming floods as stochastic processes, they can be predicted using certain probability distributions from historical streamflow data. For instance, the climatology average method (CLIM), empirical orthogonal function (EOF) , multiple linear regressions (MLR), quantile regression techniques (QRT) , and Bayesian forecasting models are widely used for predicting major floods. However, they were reported to be unsuitable for short-term prediction, and, in this context, they need major improvement due to the lack of accuracy, complexity of the usage, computation cost, and robustness of the method. Furthermore, for reliable long-term prediction, at least, a decade of data from measurement gauges should be analyzed for a meaningful forecast. In the absence of such a dataset, however, FFA can be done using hydrologic models of RFFA, e.g., MISBA and Sacramento, as reliable empirical methods with regional applications, where streamflow measurements are unavailable. In this context, distributed numerical models are used as an attractive solution. Nonetheless, they do not provide quantitative flood predictions, and their forecast skill level is “only moderate” and they lack accuracy. The drawbacks of the physically based and statistical models mentioned above encourage the usage of advanced data-driven models, e.g., machine learning (ML)

**3.1.1.DISADVANTAGES**

* Loss of livelihood & loss of life
* The existing approach has lack of accuracy.
* Computation cost is also high.

**3.2.PROPOSED SYSTEM**

Predicting the extent of calamity requires lots of parameters like rainfall, vegetation etc. Improper disposal of solid waste, including domestic, commercial and industrial waste and dumping of construction debris into the drains also contributes significantly to reducing their capacities. It is imperative to take better operations and maintenance actions. The main objective of this problem is to help the city managers or urban residents in for predicting the impact of urban flooding and can help them to take preventive action. This survey identifies the state of the art of ML methods for flood prediction where peer-reviewed articles in top-level subject fields are reviewed. Among the articles identified, through search queries using the search strategy, those including the performance evaluation and comparison of ML methods were given priority to be included in the review to identify the ML methods that perform better in particular applications. Furthermore, to choose an article, four types of quality measure for each article were considered, i.e., source normalized impact per paper (SNIP), CiteScore, SCImago journal rank (SJR), and h-index. The papers were reviewed in terms of flood resource variables, ML methods, prediction type, and the obtained results. The applications in flood prediction can be classified according to flood resource variables, i.e., water level, river flood, soil moisture, rainfall–discharge, precipitation, river inflow, peak flow, river flow, rainfall–runoff, flash flood, rainfall, streamflow, seasonal stream flow, flood peak discharge, urban flood, plain flood, groundwater level, rainfall stage, flood frequency analysis, flood quantiles, surge level, extreme flow, storm surge, typhoon rainfall, and daily flows. Among these key influencing flood resource variables, rainfall and the spatial examination of the hydrologic cycle had the most remarkable role in runoff and flood modeling. This is the reason why quantitative rainfall prediction, including avalanches, slush flow, and melting snow, is traditionally used for flood prediction, especially in the prediction of flash floods or short-term flood prediction.

However, rainfall prediction was shown to be inadequate for accurate flood prediction. For instance, the prediction of streamflow in a long-term flood prediction scenario depends on soil moisture estimates in a catchment, in addition to rainfall. Although, high-resolution precipitation forecasting is essential, other flood resource variables were considered. Thus, the methodology of this literature review aims to include the most effective flood resource variables in the search queries. A combination of these flood resource variables and ML methods was used to implement the complete list of search queries. Note that the ML methods for flood prediction may vary significantly according to the application, dataset, and prediction type. For instance, ML methods used for short-term water level prediction are significantly different from those used for long-term streamflow prediction.

**3.2.1.ADVANTAGES**

* It can help them to take preventive action.
* It improves accuracy of the model.
* Errors are reduced and correct prediction is done.

**3.3 SYSTEM REQUIREMENT SPECIFICATIONS:**

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

* Python idel 3.7 version (or)
* Anaconda 3.7 ( or)
* Jupiter (or)
* Google colab

**HARDWARE REQUIREMENTS**

Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Canopy / VS Code user. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

* Operating system : windows, linux
* Processor : minimum intel i3
* Ram : minimum 4 gb
* Hard disk : minimum 250gb

**FUNCTIONAL REQUIREMENTS**

1.Data Collection

2.Data Preprocessing

3.Training And Testing

4.Modiling

5.Predicting

### **NON FUNCTIONAL REQUIREMENTS**

NON-FUNCTIONAL REQUIREMENT (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Example of nonfunctional requirement, “how fast does the website load?”Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Non- functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are > 10000. Description of non-functional requirements is just as critical as a functional requirement.

* Usability requirement
* Serviceability requirement
* Manageability requirement
* Recoverability requirement
* Security requirement
* Data Integrity requirement
* Capacity requirement
* Availability requirement
* Scalability requirement
* Interoperability requirement
* Reliability requirement
* Maintainability requirement
* Regulatory requirement
* Environmental requirement

**SYSTEM STUDY**

**FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**SOCIAL FEASIBILITY**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system .

**3.3.1.PURPOSE**

The purpose of this document is to build a model that predicts the urban flooding in major cities. By doing this we can know the flood occurrence well in advance and can take proper measures to save the lives of people and their properties.

**3.3.2.SCOPE**

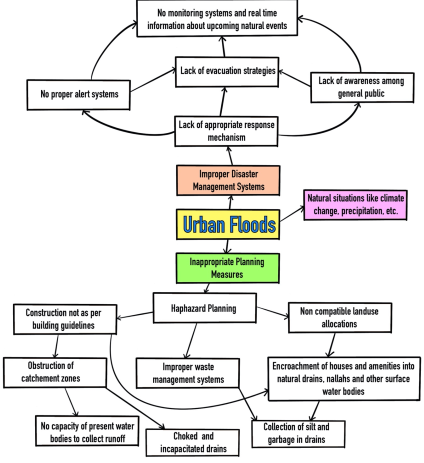
The scope of this project is to know about the floods occurrence before it occurs and alert the people and government to protect and save peoples lives. Data pre-processing, the initial part of the project is to understand implementation and usage of various python inbuilt modules. The above process helps us to understand why different modules are helpful rather than implementing those functions from scratch by the developer. These various modules provide better code representation and user understandability.

**3.3.3.OVERALL DESCRIPTION**

In this decade the upcoming technologies are mainly dependent on data. This data can only be obtained if there is some research applied on the context of the requirements of the tool. Since a lot of machine learning enthusiasts develop models which helps solve multiple problems the requirements of appropriate data are very large scale this project aims to provide a better understanding towards prediction of urban flooding in major cities. This analysis proves to be better input to machine learning models which essentially explore the rainfall dataset. These models require proper learning instances which provides better accuracy for these models. Our project ensures to provide an accurate prediction of Urban flooding in major cities.

**4.SYSTEM DESIGN**

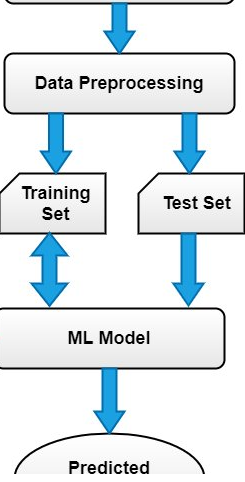
**SYSTEM ARCHITECTURE:**

****

**Fig.4.1**

**Flow Chart:**

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows.

****

**Fig.4.2**

**DATA FLOW DIAGRAM:**

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

**IMPORTING DATASET**

**VALID**

**NO PROCESS**

**Yes NO**

**DATASET PREPROCESSING**

ENTER STUDENT DATAFEATURES SELECTION

MODEL BUILDING

ACCURACY COMPARISON

**End process**

**Fig.4.3**

**4.1.UML DIAGRAMS**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

**GOALS:**

The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

**4.1.1.Use case diagram:**

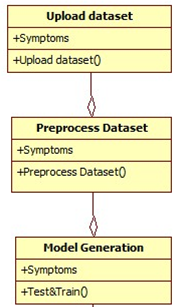
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

Fi

**Fig.4.1.1**

**4.1.2.Class diagram:**

The class diagram is used to refine the use case diagram and define a detailed design of the system. The class diagram classifies the actors defined in the use case diagram into a set of interrelated classes. The relationship or association between the classes can be either an "is-a" or "has-a" relationship. Each class in the class diagram may be capable of providing certain functionalities. These functionalities provided by the class are termed "methods" of the class. Apart from this, each class may have certain "attributes" that uniquely identify the class.



**Fig.4.1.2**

**4.1.3.Activity diagram:**

The process flows in the system are captured in the activity diagram. Similar to a state diagram, an activity diagram also consists of activities, actions, transitions, initial and final states, and guard conditions.

Data exploration

Pre-processing

Features extraction

Algorithm generation

Accuracy graph

**Fig.4.1.3**

**4.1.4.Sequence diagram:**

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".



**Fig.4.1.4**

**4.1.5.Collaboration diagram:**

A collaboration diagram groups together the interactions between different objects. The interactions are listed as numbered interactions that help to trace the sequence of the interactions. The collaboration diagram helps to identify all the possible interactions that each object has with other objects.

****

**Fig.4.1.5**

**4.1.6.Component diagram:**

The component diagram represents the high-level parts that make up the system. This diagram depicts, at a high level, what components form part of the system and how they are interrelated. A component diagram depicts the components culled after the system has undergone the development or construction phase.



**Fig.4.1.6**

**4.1.7.Deployment diagram:**

The deployment diagram captures the configuration of the runtime elements of the application. This diagram is by far most useful when a system is built and ready to be deployed.

****

**Fig.4.1.7**

**5.IMPLEMENTATION**

**5.1. MODULES**

To implement this project, I have used following modules:

1. Importing libraries.
2. Importing dataset.
3. Checking for the null values.
4. Applying label encoding to the variables.
5. Checking for the correlation among the variables.
6. Feature Selection of the variables.
7. Model Building

* SVM with gridsearchcv
* Random Forest with gridsearchcv
* Linear Discriminant Analysis
* Gaussian naive bayes with gridsearchcv
* Ensemble of logistic Regression + Gaussian Naive Bayes + Random Forest

Accuracy Comparison

**5.1.1.Modules Used in Project**

**Tensorflow**

TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software) and [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software library for dataflow and differentiable programming](https://en.wikipedia.org/wiki/Library_(computing)) across a range of tasks. It is a symbolic math library, and is also used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications such as [neural networks](https://en.wikipedia.org/wiki/Neural_networks). It is used for both research and production at [Google](https://en.wikipedia.org/wiki/Google).‍

TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain) team for internal Google use. It was released under the [Apache 2.0](https://en.wikipedia.org/wiki/Apache_License) [open-source license](https://en.wikipedia.org/wiki/Open-source_license) on November 9, 2015.

**Numpy**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

**Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Matplotlib**

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots](https://matplotlib.org/tutorials/introductory/sample_plots.html) and [thumbnail gallery](https://matplotlib.org/gallery/index.html).

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

**Scikit – learn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

**5.2.MODULE DESCRIPTION**

**RANDOM FOREST:**

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.



**SVM:**

SVM is a supervised machine learning algorithm which can be used for classification or regression problems. It uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs.



The kernel trick takes the data you give it and transforms it. In goes some great features which you think are going to make a great classifier, and out comes some data that you don't recognize anymore. It is sort of like unraveling a strand of DNA. You start with this harmelss looking vector of data and after putting it through the kernel trick, it's unraveled and compounded itself until it's now a much larger set of data that can't be understood by looking at a spreadsheet. But here lies the magic, in expanding the dataset there are now more obvious boundaries between your classes and the SVM algorithm is able to compute a much more optimal hyperplane.

**NAÏVE BAYES:**

Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

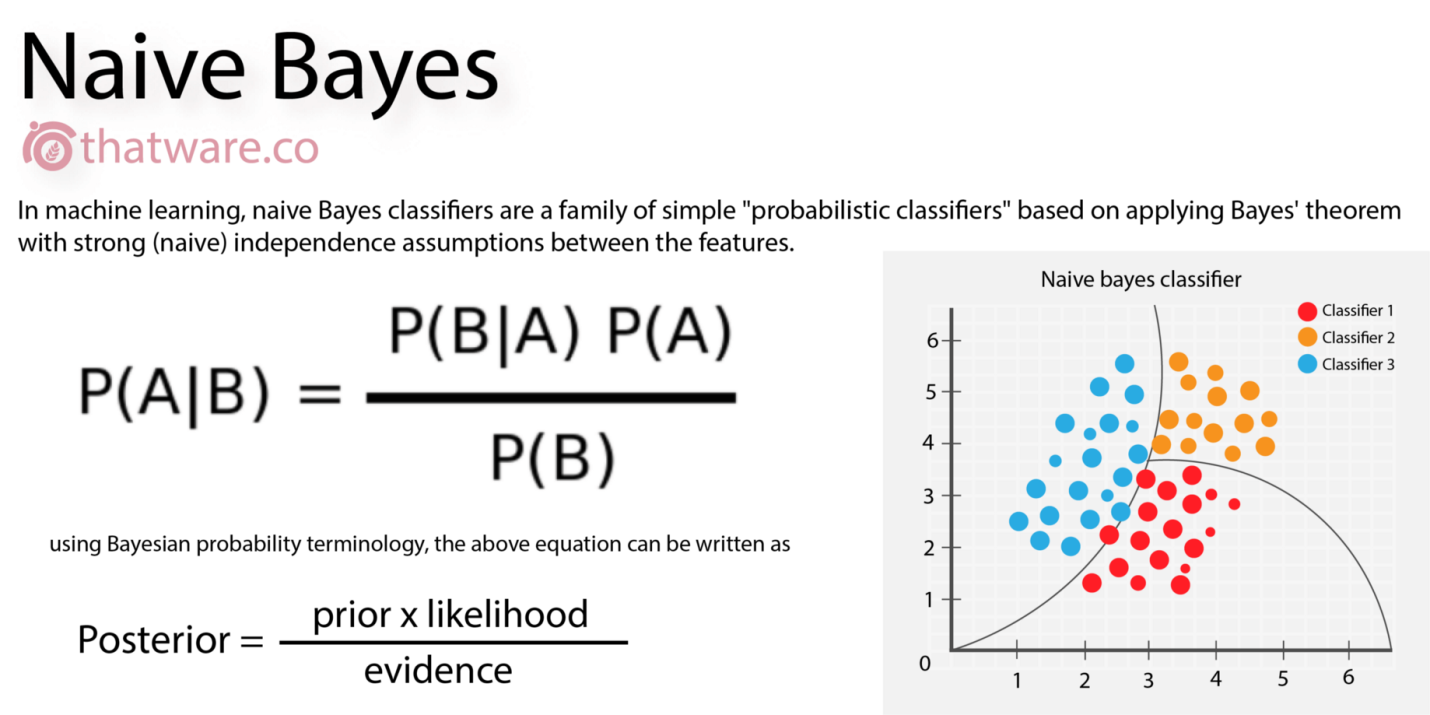
**Applications of Naive Bayes Algorithms**

Real time Prediction: Naive Bayes is an eager learning classifier and it is sure fast. Thus, it could be used for making predictions in real time.

Multi class Prediction: This algorithm is also well known for multi class prediction feature. Here we can predict the probability of multiple classes of target variable.

Text classification/ Spam Filtering/ Sentiment Analysis: Naive Bayes classifiers mostly used in text classification (due to better result in multi class problems and independence rule) have higher success rate as compared to other algorithms. As a result, it is widely used in Spam filtering (identify spam e-mail) and Sentiment Analysis (in social media analysis, to identify positive and negative customer sentiments)

Recommendation System: Naive Bayes Classifier and Collaborative Filtering together builds a Recommendation System that uses machine learning and data mining techniques to filter unseen information and predict whether a user would like a given resource or not.



**5.3.INTRODUCTION TO TECHNOLOGIES USED**

# 5.3.1.What is Python

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* + [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
  + GUI Applications (like Kivy, Tkinter, PyQt etc. )
  + Web frameworks like Django (used by YouTube, Instagram, Dropbox)
  + Image processing (like Opencv, Pillow)
  + Web scraping (like Scrapy, BeautifulSoup, Selenium)
  + Test frameworks
  + Multimedia

### Advantages of Python :-

Let’s see how Python dominates over other languages.

#### 1. Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

#### 2. Extensible

As we have seen earlier, Python can be**extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

#### 3. Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities**to our code in the other language.

#### 4. Improved Productivity

The language’s simplicity and extensive libraries render programmers**more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

#### 5. IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

#### 6. Simple and Easy

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and**code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

#### 7. Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

#### 8. Object-Oriented

This language supports both the **procedural and object-oriented**programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

#### 9. Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you[**download Python**](https://data-flair.training/blogs/install-python-windows/) for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

#### 10. Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to**code only once**, and you can run it anywhere. This is called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

#### 11. Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

### **5.3.2.Advantages of Python Over Other Languages**

#### 1. Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

#### 2. Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

#### 3. Python is for Everyone

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and [**machine learning**](https://data-flair.training/blogs/machine-learning-tutorials-home/), automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

### **5.3.3.Disadvantages of Python**

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

#### 1. Speed Limitations

We have seen that Python code is executed line by line. But since [Python](https://www.python.org/) is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

#### 2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

#### 3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can**raise run-time errors**.

#### 4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

#### 5. Simple

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

**5.3.4.History of Python : -**

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

**5.3.5.What is Machine Learning : -**

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

**5.3.6.Categories Of Machine Leaning :-**

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

## 5.3.7.Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

## 5.3.8.Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

**Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

**Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

**Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

**No clear objective for formulating business problems** − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

**Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

## 5.3.9.Applications of Machines Learning :-

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

* Emotion analysis
* Sentiment analysis
* Error detection and prevention
* Weather forecasting and prediction
* Stock market analysis and forecasting
* Speech synthesis
* Speech recognition
* Customer segmentation
* Object recognition
* Fraud detection
* Fraud prevention
* Recommendation of products to customer in online shopping

# 5.3.10.How to Start Learning Machine Learning?

Arthur Samuel coined the term **“Machine Learning”** in 1959 and defined it as a **“Field of study that gives computers the capability to learn without being explicitly programmed”.**

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to [Indeed](http://blog.indeed.com/2019/03/14/best-jobs-2019/), Machine Learning Engineer Is The Best Job of 2019 with a 344% growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

### **How to start learning ML?**

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

### Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

#### (a) Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

#### (b) Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!  
Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

#### (c) Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is [Python](https://www.geeksforgeeks.org/python-programming-language/)! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as [Keras](https://keras.io/" \t "_blank), [TensorFlow](https://www.tensorflow.org/), [Scikit-learn](https://scikit-learn.org/stable/), etc.

So if you want to learn ML, it’s best if you learn Python! You can do that using various online resources and courses such as [**Fork Python**](https://practice.geeksforgeeks.org/courses/fork-python) available Free on GeeksforGeeks.

### **Step 2 – Learn Various ML Concepts**

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It’s best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

#### (a) Terminologies of Machine Learning

* **Model –**A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
* **Feature –**A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
* **Target (Label) –**A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
* **Training –**The idea is to give a set of inputs(features) and it’s expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
* **Prediction –**Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

#### (b) Types of Machine Learning

* **Supervised Learning –**This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
* **Unsupervised Learning –**This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
* **Semi-supervised Learning –**This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
* **Reinforcement Learning –**This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

### **5.3.11.Advantages of Machine learning :-**

#### 1. Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

#### 2. No human intervention needed (automation)

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

#### 3. Continuous Improvement

As [**ML algorithms**](https://data-flair.training/blogs/machine-learning-algorithms/) gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

#### 4. Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

#### 5. Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

### **5.3.12.Disadvantages of Machine Learning :-**

#### 1. Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

#### 2. Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

#### 3. Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

#### 4. High error-susceptibility

[**Machine Learning**](https://en.wikipedia.org/wiki/Machine_learning) is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

**5.3.13.Python Development Steps : -**

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system.  
Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

* Print is now a function
* Views and iterators instead of lists
* The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
* There is only one integer type left, i.e. int. long is int as well.
* The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
* Text Vs. Data Instead Of Unicode Vs. 8-bit

**5.3.14.Purpose :-**

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

**Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Install Python Step-by-Step in Windows and Mac :**

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

## How to Install Python on Windows and Mac :

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

**Note:** The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. [Download the Python Cheatsheet here.](https://myelearninghub.com/python-cheat-sheet/)The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

### Download the Correct version into the system

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: [**https://www.python.org**](https://www.python.org/)



Now, check for the latest and the correct version for your operating system.

**Step 2:** Click on the Download Tab.

****

**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

****

**Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.



• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.

•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

### Installation of Python

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.



**Step 2:** Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes.

### Verify the Python Installation

**Step 1:** Click on Start

**Step 2:** In the Windows Run Command, type “cmd”.



**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python –V** and press Enter.



**Step 5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

### Check how the Python IDLE works

**Step 1:** Click on Start

**Step 2:** In the Windows Run command, type “python idle”.



**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**Step 4:** To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



**Step 5:** Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

**Step 6:** Now for e.g. **enter print**.

**5.4.SAMPLE CODE:**

**#Importing Libraries**

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import warnings

warnings.filterwarnings('ignore')

import plotly.figure\_factory as ff

import plotly.express as px

from sklearn.model\_selection import GridSearchCV

from sklearn import preprocessing

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import confusion\_matrix, accuracy\_score, classification\_report

from sklearn import metrics

import pickle

**#Importing Dataset**

dataset = pd.read\_csv("Dataset/indian rainfall.csv")

**#EDA**

dataset.head()

dataset.info()

dataset.isnull().sum()

dataset['SUBDIVISION'].unique()

**#Data Visualization**

sns.countplot(data=dataset, x='FLOOD')

corrmat = dataset.corr()

plt.subplots(figsize=(18,18))

sns.heatmap(corrmat,annot=True, square=True, vmin=0, vmax=1,cmap="YlGnBu");

**#Feature selection using correlation**

label\_encoder = preprocessing.LabelEncoder()

dataset['FLOOD']= label\_encoder.fit\_transform(dataset['FLOOD'])

dataset['SUBDIVISION']= label\_encoder.fit\_transform(dataset['SUBDIVISION'])

dataset

print(dataset.corr()["FLOOD"].abs().sort\_values(ascending=False))

X = dataset[['ANNUAL','Jun-Sep','JUN','JUL','AUG','SEP','MAY','Mar-May','OCT','Oct-Dec']]

y = dataset['FLOOD']

**#Model Building**

x\_train, x\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=0)

from sklearn.metrics import confusion\_matrix, accuracy\_score, classification\_report

from sklearn import metrics

def evaluate(model, X\_train, X\_test, y\_train, y\_test):

y\_test\_pred = model.predict(X\_test)

y\_train\_pred = model.predict(X\_train)

print("TRAINIG RESULTS: \n===============================")

clf\_report = pd.DataFrame(classification\_report(y\_train, y\_train\_pred, output\_dict=True))

print(f"CONFUSION MATRIX:\n{confusion\_matrix(y\_train, y\_train\_pred)}")

print(f"ACCURACY SCORE:\n{accuracy\_score(y\_train, y\_train\_pred):.4f}")

print(f"CLASSIFICATION REPORT:\n{clf\_report}")

print("TESTING RESULTS: \n===============================")

clf\_report = pd.DataFrame(classification\_report(y\_test, y\_test\_pred, output\_dict=True))

print(f"CONFUSION MATRIX:\n{confusion\_matrix(y\_test, y\_test\_pred)}")

print(f"ACCURACY SCORE:\n{accuracy\_score(y\_test, y\_test\_pred):.4f}")

print(f"CLASSIFICATION REPORT:\n{clf\_report}")

from sklearn.ensemble import RandomForestClassifier

RT = RandomForestClassifier()

param\_grid = {

'n\_estimators': [150, 450],

'max\_features': ['auto', 'sqrt', 'log2'],

'max\_depth' : [4,5,6,7,8],

'criterion' :['gini', 'entropy']

}

RT = GridSearchCV(estimator=RT, param\_grid=param\_grid, cv= 5)

RT.fit(x\_train, y\_train)

predictions = RT.predict(x\_test)

val2 = (accuracy\_score(y\_test, predictions)\*100)

evaluate(RT, x\_train, x\_test, y\_train, y\_test)

**#Pickle**

s = np.array([3369,1698.36,562,369,458.3,332.6,456.39,569.356,458.36,78.9])

print(s.shape)

s = s.reshape(1,-1)

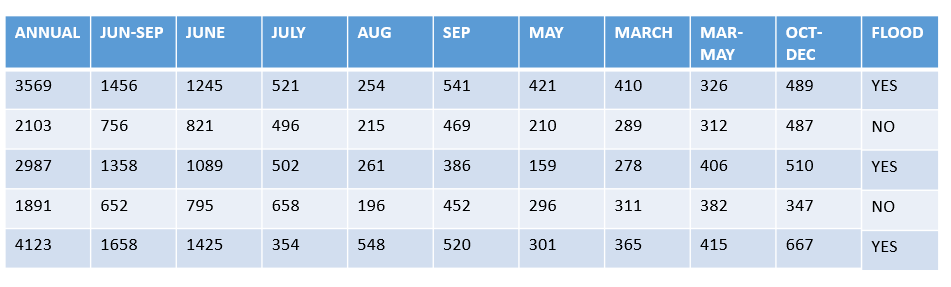
print(s.shape)

pickle.dump(RT, open('model.pkl', 'wb'))

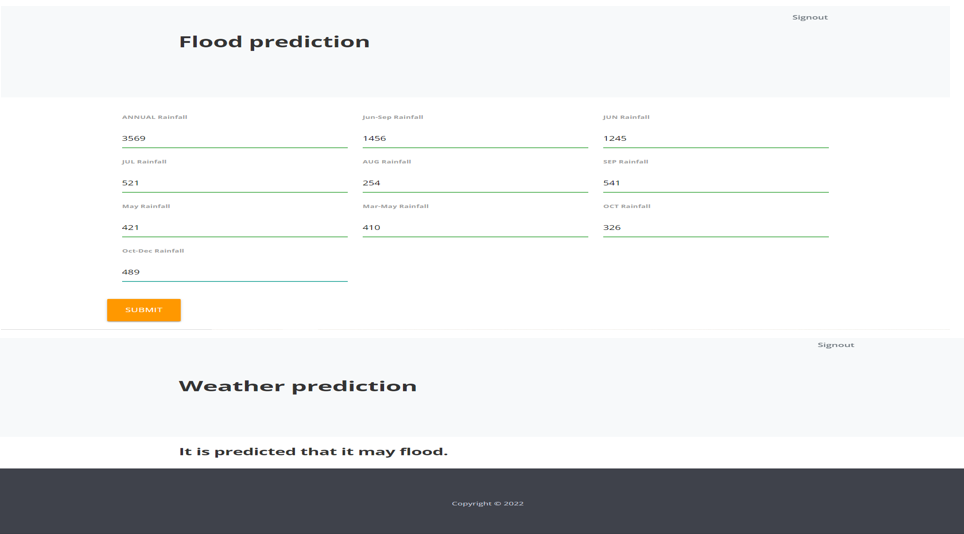
model = pickle.load(open('model.pkl', 'rb'))

print(model.predict(s))

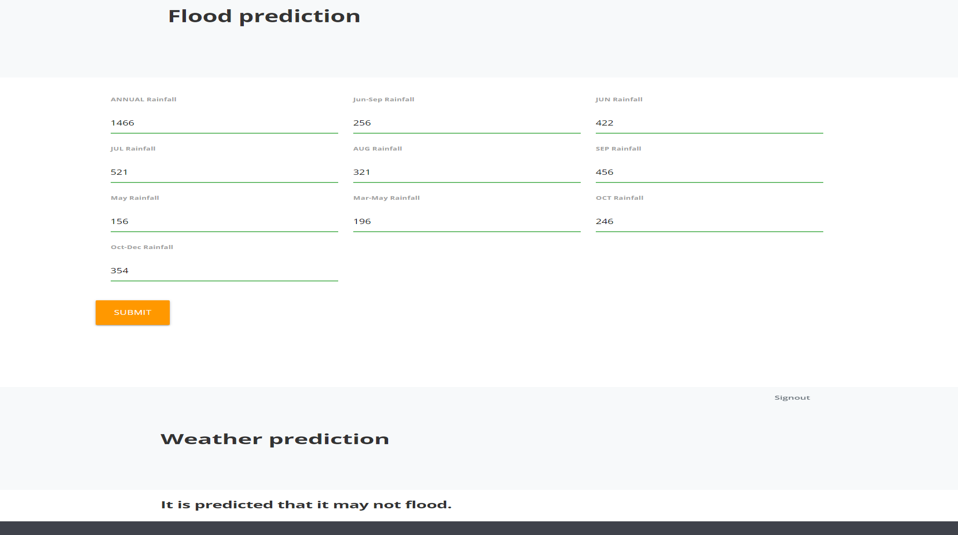
**6. TEST CASES**



**Table.6.1**

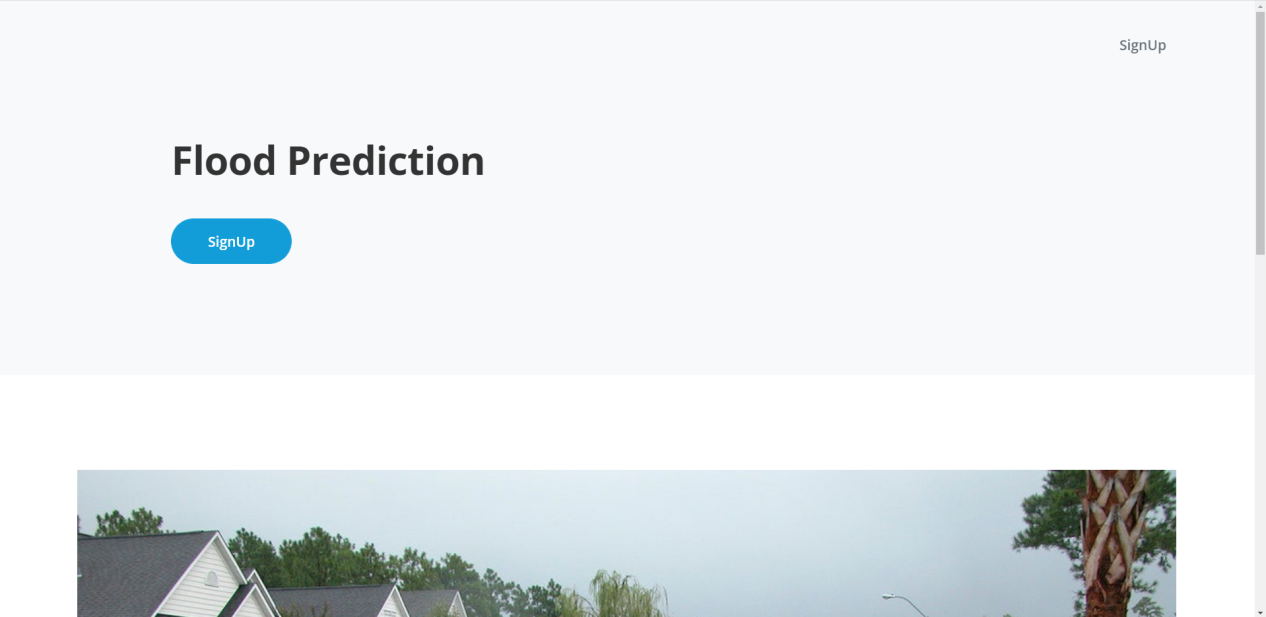
****

**Fig.6.2**

****

**Fig.6.3**

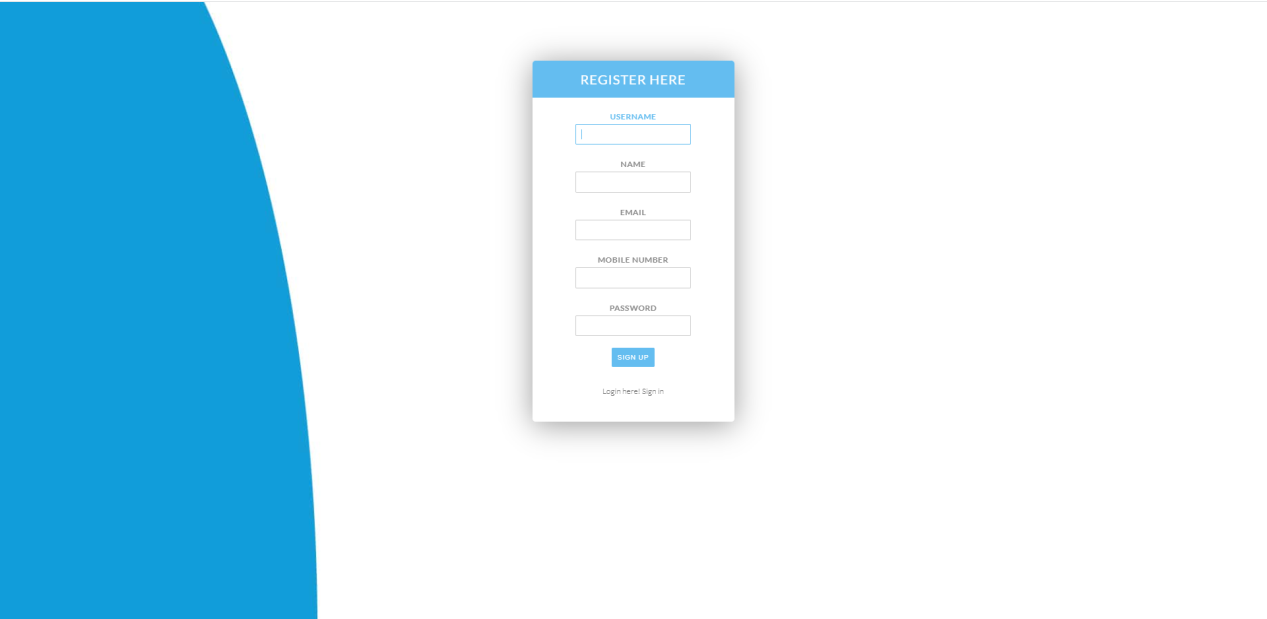
**7.SCREENSHOTS**



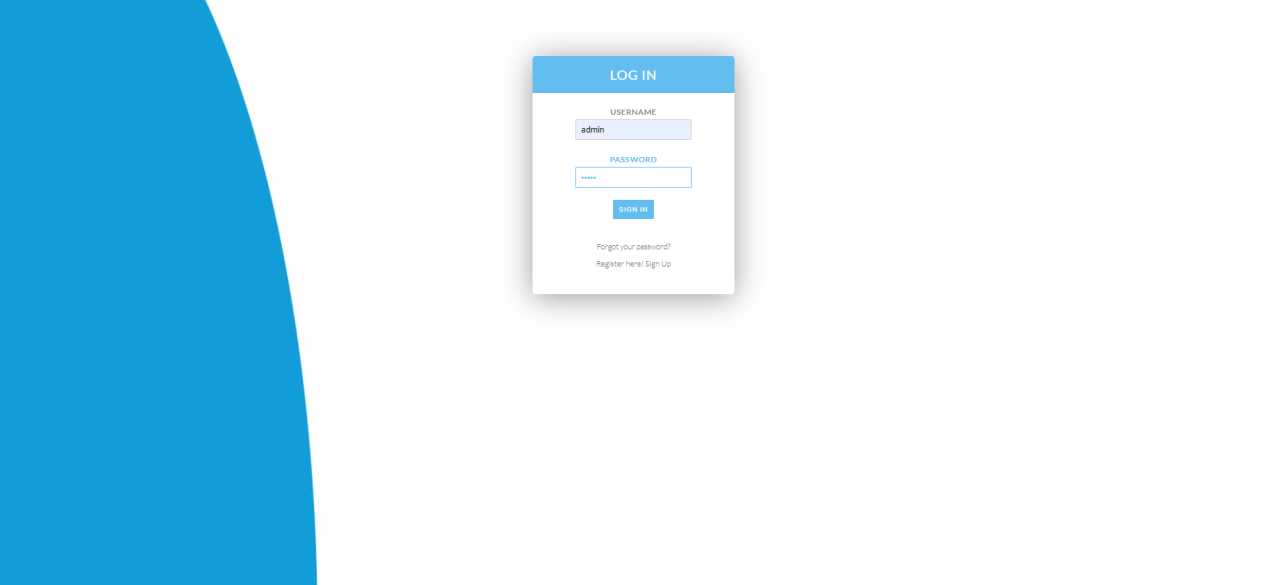
**Fig.7.1**



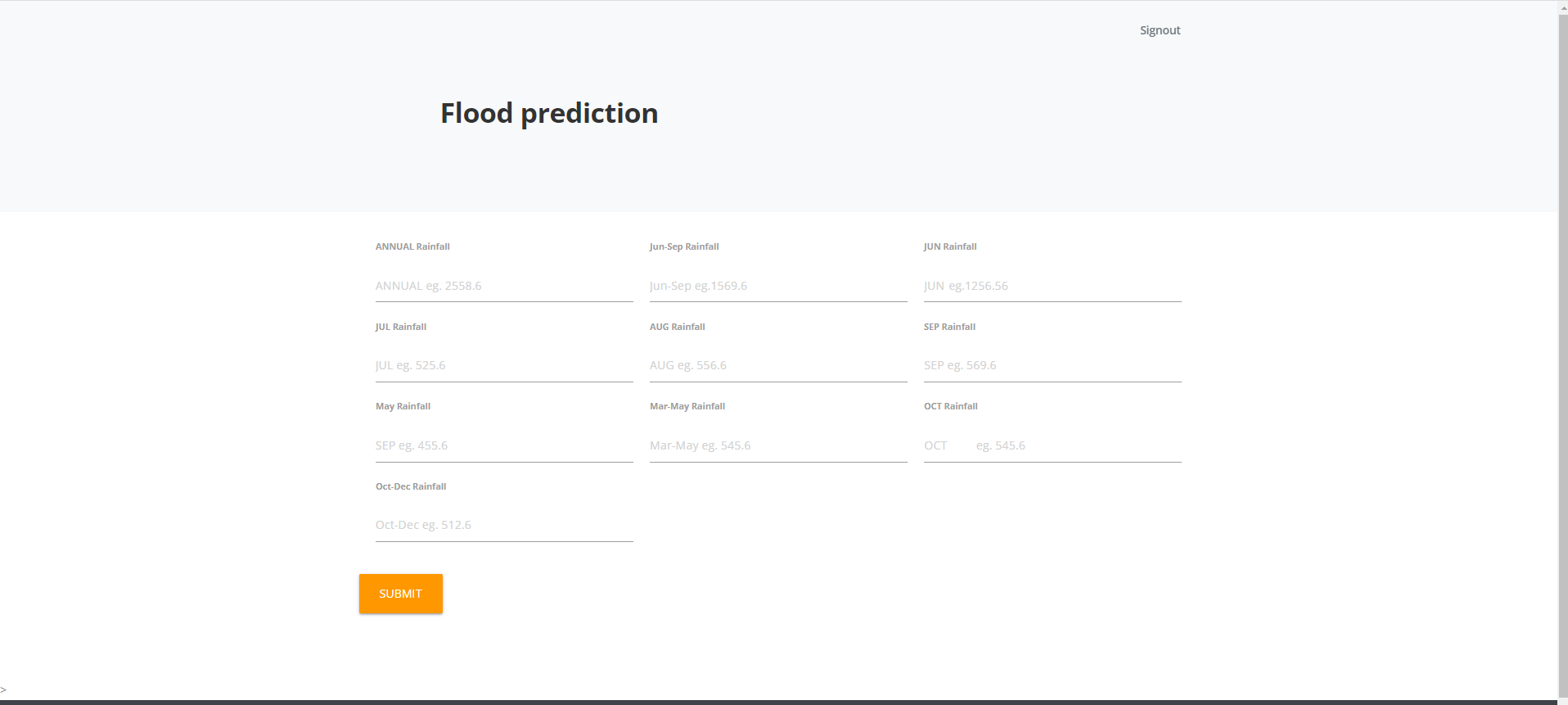
**Fig.7.2**



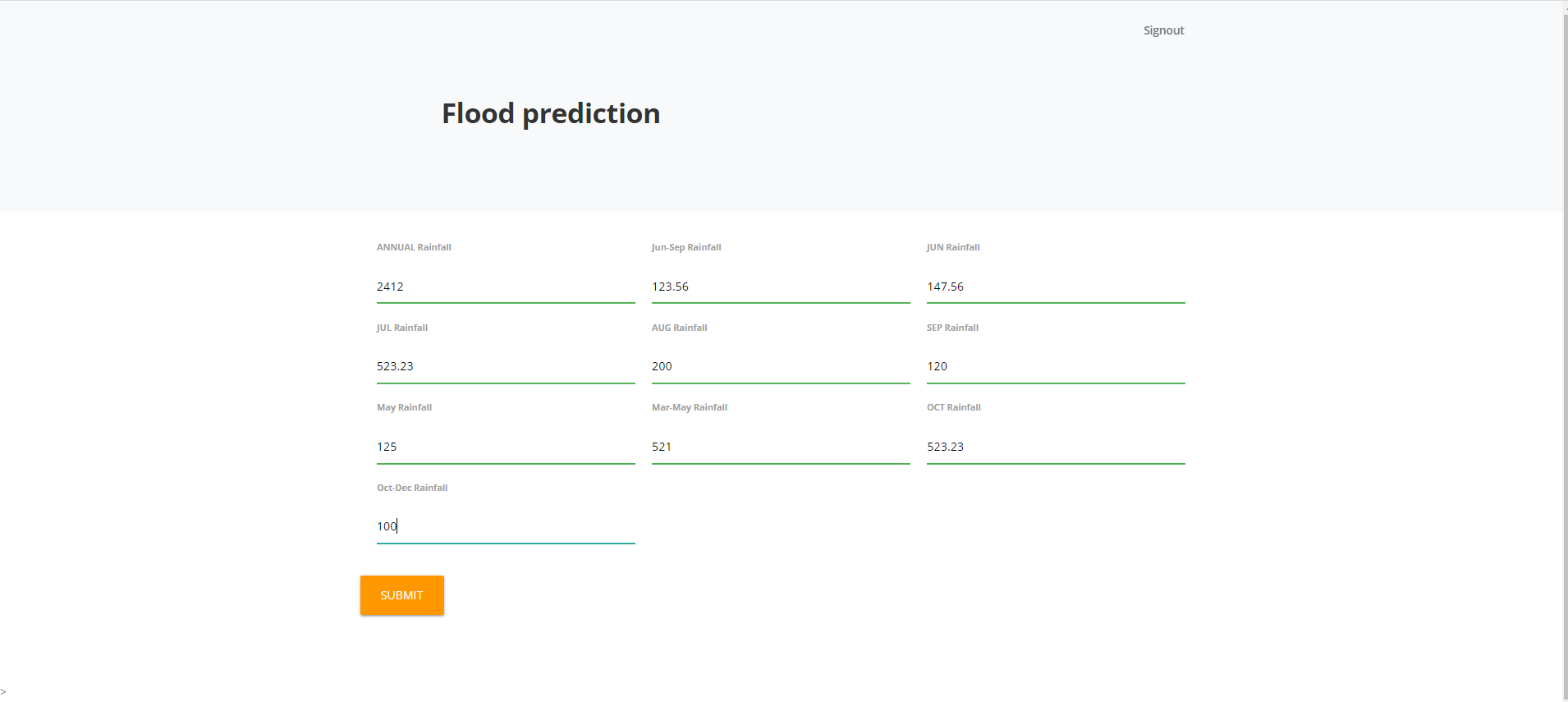
**Fig.7.3**



**Fig.7.4**



**Fig.7.5**



**Fig.7.6**

**8.CONCLUSION**

The major part of the study focuses on identifying the drivers that cause inundations in an Urban area in relation to the rapid pace of urbanization in India. Urban flooding is caused mostly due to human interventions than by natural causes. Even though changing climate has a huge role to play in increasing the hazard, climate change is an indirect result of man destroying the environment. It has been observed that due to population expansion and change in land use types, a large increase in surface runoff can be induced. Analysing the causes, it has been observed that the main focus for urban flood mitigation should be runoff reduction and keeping the flood plains free from obstruction. The loss of surface water bodies is directly linked to increasing the vulnerability of the settlement, as it gets more urbanized. To make better planning decisions, policymakers need to understand the sensitivity of the natural drainage pattern and topography. Assessing the current and future urban drainage in coping with the increasing risk of urban floods created by regional and local factors should be the primary concern. The impacts of urban flooding can be minimized only by making changes to the way we plan our cities. Due to increased rate of urbanization lot of people are moving to urban places from rural areas which includes lot of investment. If a flood occurs suddenly, then their properties and lives will be damaged. As a result, there is a need to protect them.So, we have planned to develop a model that predicts the urban flooding in major cities.By this project we can estimate the floods and its impact. As a result it will be helpful for both the government and people in cities.

**9.FUTURE ENHANCEMENT**

From the review outcomes discussed, we can see that although the overall number of studies on ensemble flood fore-casting has increased significantly in the last decade, this increase is not uniform across the different categories reviewed. For example, it varies with the geographic focus of the study, types of floods forecast, and ensemble modelling and evaluation methods used. This is partially due to the quality and availability of observational data. It is also because

ensemble forecasting is a relatively new concept in flood-related management applications, although it has been are search focus over the past two to three decades in hydrometeorological modelling. In the following section, we summarize some of the areas for future focus and consideration in the ensemble flood forecasting research and practice communities, including:

Improving data quality and availability

**1.Improving data quality and availability**

All forecasts, whether ensemble or deterministic, are dependent on observational data of high quality for all parts of the flood forecasting chain. Data acquisition for forecasts has undergone rapid and fundamental changes in recent decades with the rise of remote sensing and new types of data, but has at the same time confronted a decline of traditional observations (Dutra et al., 2014). This poses new opportunities as well as new challenges for the forecasting systems with regards to having the best possible input for hydrometeorological forecasting.

Catering for different flood events

**2.Catering for different flood events**

The majority of flood forecasting systems reviewed here predict river flows within defined river channels. Systemseither display little hydrological regulation (Liechti et al., 2013) or neglect the effects of regulation times of flooding(Ushiyama et al., 2014; Yu et al., 2018). The studies generally focus on meso-scale catchments where the interactionsbetween antecedent catchment conditions and forecast weather are both important and there are multiple different methods of producing forecasts with similar performance.

•Improving data quality and availability,

•Catering for different types of flood events,

•Extending flood forecast variables,

•Extending and bridging the gap in flood forecast lead time, and

•Extending ensemble generation and evaluation.Finally, we suggest potential research directions in order to move forward and to encourage the uptake of ensemble

Improving data quality and availability,

•Catering for different types of flood events,

•Extending flood forecast variables,

•Extending and bridging the gap in flood forecast lead time, and

•Extending ensemble generation and evaluation.

**3.Extending Flood Forecast Variables**

The majority of flood forecasting systems reviewed here predict river flow as their main output variable, yet there arean increasing number of studies describing models predicting other flood variables, such as inundation and water levels. Typically, producing forecasts of inundation and water levels requires the use of a hydraulic model in addition toa rainfall-runoff model.

**BIBLIOGRAPHY**

1. Stevens AJ, Clarke D, Nicholls RJ. 2014 Trends in reported flooding in the UK: 1884–2013. Hydrol. Sci. J. 61, 50–63. (doi:10.1080/02626667.2014.950581)

2. Escarameia M, Karanxha A, Tagg A. 2007 Quantifying the flood resilience properties of walls in typical UK dwellings. Build. Serv. Eng. Res. Technol. 28, 249–263. (doi:10.1177/0143624407079093)

3. Golz S, Schinke R, Naumann T. 2015 Assessing the effects of flood resilience technologies on building scale. Urban Water J.12, 30–43. (doi:10.1080/ 1573062X.2014.939090)

4. Handmer J, Proudley B. 2007 Communicating uncertainty via probabilities: the case of weather forecasts.Environ. Hazards7, 79–87. (doi:10.1016/j. envhaz.2007.05.002)

5. Huurne ET, Gutteling J. 2008 Information needs and risk perception as predictors of risk information seeking. J. Risk Res.11, 847–862. (doi:10.1080/ 13669870701875750)

6. Krohne HW. 1989 The concept of coping modes: relating cognitive person variables to actual coping behaviour. Adv. Behav. Res. Ther.11, 235–249. (doi:10.1016/0146-6402(89)90027-1)

7. Zuuren FJ, Wolfs HM. 1991 Styles of information seeking under threat: personal and situational aspects of monitoring and blunting.Pers. Indiv. Differ.12, 141–149. (doi:10.1016/0191-8869(91) 90097-U)

8. Falagas ME, Karveli EA, Tritsaroli VI. 2008 The risk of using the Internet as reference resource: a comparative study. Int. J. Med. Inform.77, 280–286. (doi:10.1016/j.ijmedinf.2007.07.001)

9. Baltierra NB, Muessig KE, Pike EC, LeGrand S, Bull SS, Hightow-Weidman LB. 2016 More than just tracking time: complex measures of user engagement with an internet-based health promotion intervention. J. Biomed. Inform.59, 299–307. (doi:10.1016/j.jbi.2015.12.015)

10. Kuo MS, Chuang TY. 2016 How gamification motivates visits and engagement for online academic dissemination—an empirical study. Comput. Hum. Behav.55, 16–27. (doi:10.1016/j. chb.2015.08.025)