**Machine Learning based Rainfall Prediction**

**ABSTRACT :**

Rainfall prediction is one of the challenging and uncertain tasks which has a significant impact on human society. Timely and accurate predictions can help to proactively reduce human and financial loss. Rainfall prediction has a significant impact in the world. It plays a crucial role in numerous sectors including economy and especially in agriculture sector. Prediction of rainfall is one of the challenging task as it can help us undertake various measures to prevent situations which are prone to danger during natural disasters, thus minimizes the risk of danger. Prediction of short term rainfall is done and there is a possibility of uncertainity. Also, complexity of datasets can increase which may lead to difficulty. While for long term rainfall, heavy precipitation is causing a drawback to the society. Error rates are a major factor that is considered when large datasets are being used. High error rates may again result in uncertain data. To overcome this drawback machine learning techniques can be used for accurate precipitation prediction. As already known, machine learning is the subset of Artificial intelligence and consist of different types of learning. By eliminating uncertainity these techniques give accurate and timely prediction. Some studies of machine learning techniques like classification is being used in this process of prediction. Wide range of algorithms can be used in the process of prediction but each of them have different amount of accuracy rates. This study presents a set of experiments which involve the use of prevalent machine learning techniques to build models to predict whether it is going to rain tomorrow or not based on weather data for that particular day in major cities of Australia. This comparative study is conducted concentrating on three aspects: modeling inputs, modeling methods, and pre-processing techniques. The results provide a comparison of various evaluation metrics of these machine learning techniques and their reliability to predict the rainfall by analyzing the weather data.

**INTRODUCTION :**

India’s welfare is agriculture. The achievement of agriculture is dependent on rainfall. It also

helps with water resources. Rainfall information in the past helps farmers better manage their

crops, leading to economic growth in the country. Prediction of precipitation is beneficial to

prevent flooding that saves people's lives and property. Fluctuation in the timing of

precipitation and its amount makes forecasting of rainfall a problem for meteorological scientists. Forecasting is one of the utmost challenges for researchers from a variety of fields, such as weather data mining, environmental machine learning, functional hydrology, and numerical forecasting, to create a predictive model for accurate rainfall. In these problems, a common question is how to infer the past predictions and make use of future predictions. A variety of sub-processes are typically composed of the substantial process in rainfall. It is at times not promising to predict the precipitation correctly by on its global system. Climate forecasting stands out for all countries around the globe in all the benefits and services provided by the meteorological department. The job is very complicated because it needs specific numbers and all signals are intimated without any assurance. Accurate precipitation forecasting has been an important issue in hydrological science as early notice of stern weather can help avoid natural disaster injuries and damage if prompt and accurate forecasts are made. The theory of the modular model and the integrati2on of different models has recently gained more interest in rainfall forecasting to address this challenge. A huge range of rainfall prediction methodologies is available in India. In India, there are two primary methods of forecasting rainfall. Regression, Artificial Neural Network (ANN), Decision Tree algorithm, Fuzzy logic and team process of data handling are the majority frequently used computational methods used for weather forecasting The basic goal is to follow information rules and relationships while gaining intangible and potentially expensive knowledge. Artificial NN is a promising part of this wide field. Since the presence of current techniques which do not possess accurate results and as well as a difficulty in finding the hidden patterns or features in the data, the proposed system uses the machine learning techniques that possess higher accuracy results. The techniques that are being used are going through some steps of procedure which include importing rainfall dataset, as contains parameters from meteorological department. And the very next step includes handling amount of missing data. This missing data refers to the data which is missing from the imported dataset. And this missing data results in error rates. The amount of error rates play an important role in defining the accuracy of the ML techniques that are used. Each algorithm in Machine Learning has some percentage of overcoming the error rates. This percentage must be seen while considering an algorithm to predict the rainfall. After this step, the third step that is done is feature scaling. This feature scaling is the process of normalization. This step plays a crucial part in using the algorithms of machine learning. There is also standardization which involves in the process of feature scaling. As the dataset is imported, which contains wide range of features like atmospheric pressure, temperature, humidity etc, where in these feature values are scaled in the third step. The following step is the step of Split done between training data and testing data. This is done in such a way that maximum amount of data is sent for training and minimum amount of data is sent under testing for evaluation, This process done for almost every state of prediction. Here the model is trained and is being sent for testing. Firstly, various input parameters are taken for the process of testing, and then the data is thoroughly trained. The maximum amount of input data is 80% of data. Then it is about 20% of data is sent for testing. There are various types of errors that are possessed during the process of data exploration. These errors can be of types : RE and RMSE. These errors are abbreviated as Root errors and Root mean square errors which generally occur with every imported data.

Now, these errors are handled in such a way that the errors are reduced. Also, quality of dataset is very much prominent. This quality can define the whole process of prediction. The quality of dataset that is used here is the influences the whole process. This process is also known as decomposition technique. This technique is the one that connects with the quality of dataset being used. Then comes the way of using ensemble learning method which defines the process of standard way of predicting future result with respect to the current data or historical data. The past data in the dataset plays a major role as it includes numerous amounts of parameters which boosts the techniques or algorithms that are used in the field of prediction. With respect to the consideration of climate change which is the main highlight when it comes to the point of Global warming. Global warming is referred as a threat to the whole planet. It is a sudden rise in the temperature which has heat highly, and it deals with the issues of levels of CO2. This issue can only have chances of prevention only when there is right way of predicting rainfall.

Rainfall prediction remains a serious concern and has attracted the attention of governments, industries, risk management entities, as well as the scientific community. Rainfall is a climatic factor that affects many human activities like agricultural production, construction, power generation, forestry and tourism, among others

[1]. To this extent, rainfall prediction is essential since this variable is the one with the highest correlation with adverse natural events such as landslides, flooding, mass movements and avalanches. These incidents have affected society for years

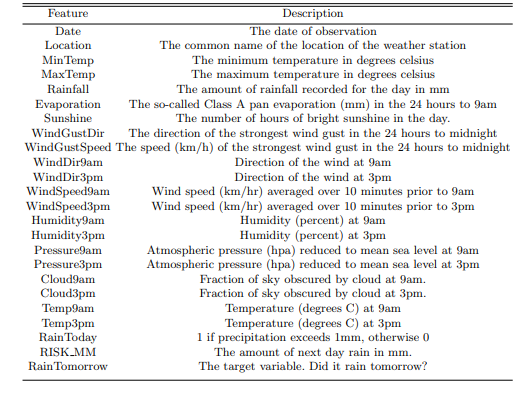
[2]. Therefore, having an appropriate approach for rainfall prediction makes it possible to take preventive and mitigation measures for these natural phenomena .

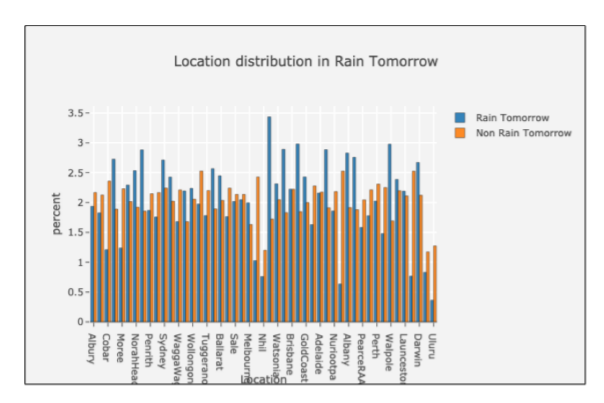
To solve this uncertainty, we used various machine learning techniques and models to make accurate and timely predictions. These paper aims to provide end to end machine learning life cycle right from Data preprocessing to implementing models to evaluating them. Data Preprocessing steps include imputing missing values, feature transformation, encoding categorical features, feature scaling and feature selection. We implemented models such as Bagging classifier and Xgboost for higher accuracies and for evaluation purpose.

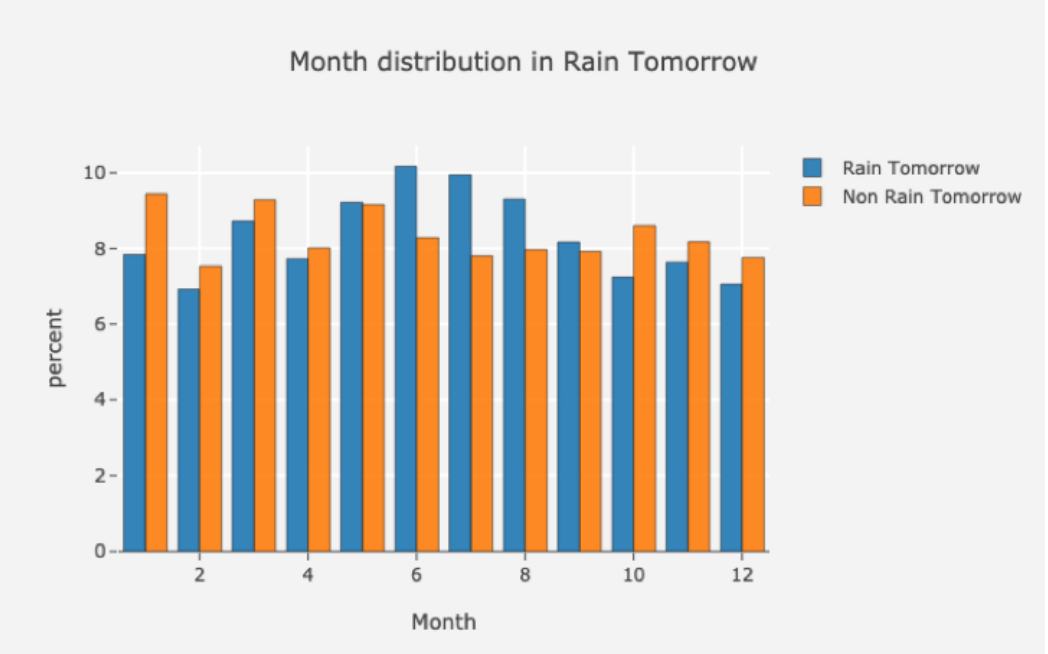
**CASE STUDY :**

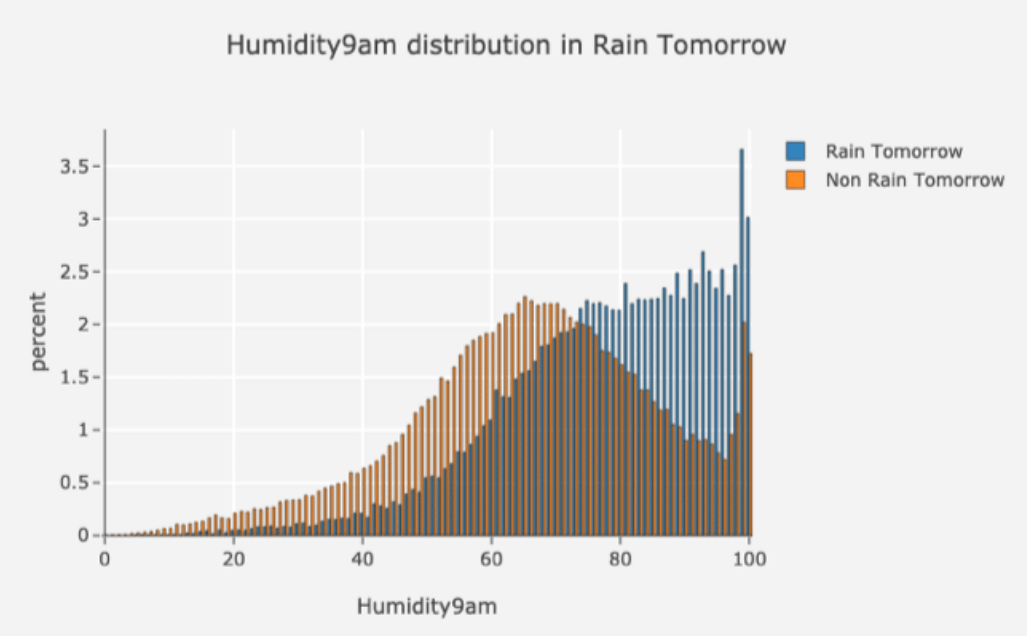
In this paper, the data set under consideration contains daily weather observations from numerous Australian weather stations. The target variable is RainTomorrow which means: Did it rain the next day? Yes or No. The dataset is available at https://www.kaggle.com/jsphyg/weather-dataset-rattle-package and definitions are adapted from http://www.bom.gov.au/climate/dwo/IDCJDW0000. shtml.

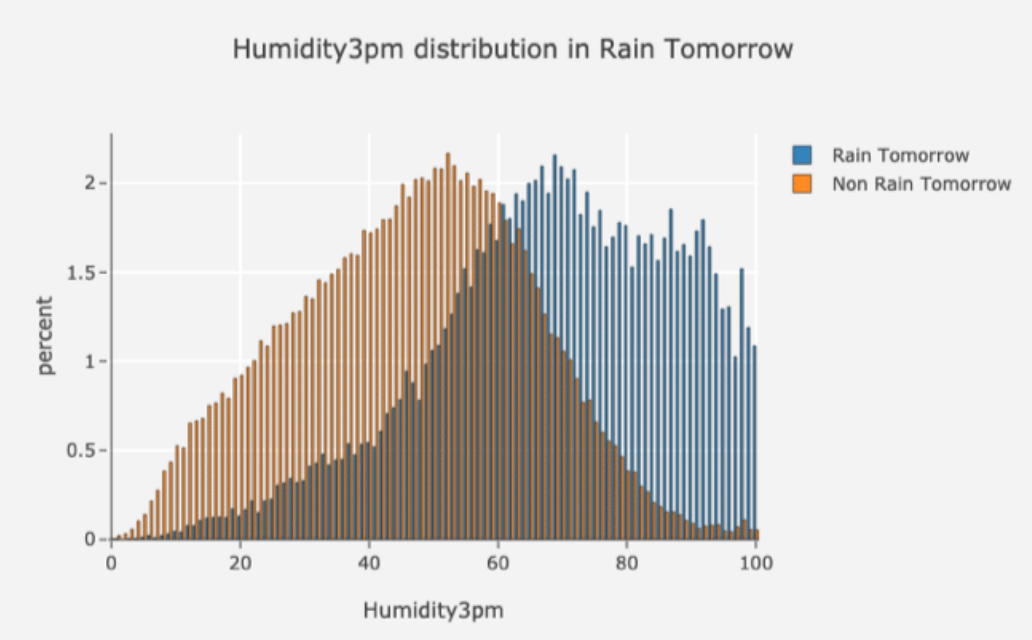
The data set consists of 23 features and 142k instances. Below are the features.

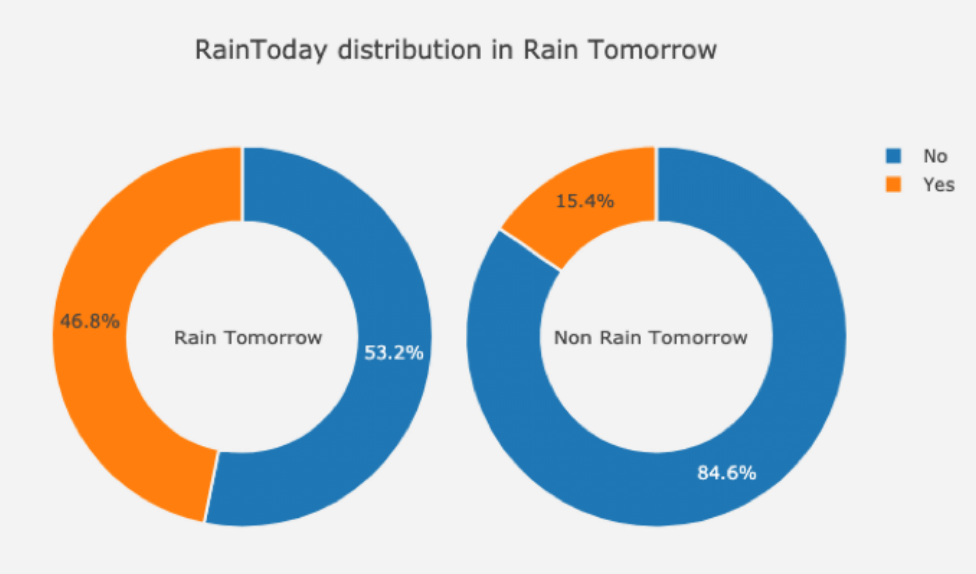












**EXISTING SYSTEM :**

Rainfall prediction is important as heavy rainfall can lead to many disasters. The prediction helps people to take preventive measures and moreover the prediction should be accurate. There are two types of prediction short term rainfall prediction and long term rainfall. Prediction mostly short term prediction can gives us the accurate result. The main challenge is to build a model for long term rainfall prediction. Heavy precipitation prediction could be a major drawback for earth science department because it is closely associated with the economy and lifetime of human.

**DISADVANTAGES :**

1. Existing system is failing whenever massive amounts of datasets are been given. Whenever this complexity of datasets increase, there is vast percentage of chances of failure of algorithm.

2. There are various existing systems which contains different algorithms that are being used. We can just do it by having the historical data analysis of rainfall and can predict the rainfall for future seasons. We can apply many techniques like classification, regression according to the requirements and also we can calculate the error between the actual and prediction and also the accuracy.

3. The approaches which are being used now-a-days are statistical methods and numerical methods, which don’t work accurately when there is any non-linear pattern like in linear regression which can’t catch the non-linearity in data.

4. Different techniques produce different accuracies so it is important to choose the right algorithm and model it according to the requirements. Complexity of datasets and error rates are highly in increase when using an existing system. Hence, this study provides efficient accurate prediction of rainfall.

**PROPOSED SYSTEM :**

It’s a cause for natural disasters like flood and drought that square measure encountered by individuals across the world each year. Accuracy of rainfall statement has nice importance for countries like India whose economy is basically dependent on agriculture. The dynamic nature of atmosphere, applied mathematics techniques fail to provide sensible accuracy for precipitation statement. The prediction of precipitation using machine learning techniques may use regression. Intention of this project is to offer non-experts easy access to the techniques, approaches utilized in the sector of precipitation prediction and provide a comparative study among the various machine learning techniques.

**ADVANTAGES :**

1. The proposed approach is to develop ML models for accurate and timely prediction of rainfall through classification algorithms that are under Supervised Learning.
2. This approach uses **Ensemble Techniques** which deals with consideration of multiple models to improve overall performance of the model. The approach can be used for Large datasets. This inturn can prevent overfitting of data.
3. Accuracy depends on the dataset. The accuracy, estimation and precision of considered algorithms / classifiers/ predictors are way better than the ones in the existing system. •
4. The proposed system provides an effective way of handling missing data with low computational cost.
5. Every feature that is present in the dataset is properly scaled by using efficient techniques.
6. To obtain high level of accuracy in prediction especially for long-term rainfall , training 80% of is being done while using efficient machine learning techniques.
7. Can reduce error rates which boosts up for higher accuracy. It is a powerful technique for testing relationship between one dependent variable and many independent variables.
8. It allows researchers to control extraneous factors.
9. Regression asses the cumulative effect of multiple factors.
10. It also helps to attain the measure of error using the regression line as a base for estimations.
11. Finally, we’re considering several ensemble techniques and a comparative study is also being done to check the overall accuracy and the performance of the model.

**LITERATURE SURVEY:**

We have surveyed and we’re going to present papers that have been studied for Rainfall Prediction using ML techniques :

[1] .Climate Change and Human Health: Risks and Responses

The long-term good health of populations depends on the continued stability and functioning of the biosphere’s ecological and physical systems, often referred to as life-support systems. We ignore this long-established historical truth at our peril: yet it is all too easy to overlook this dependency, particularly at a time when the human species is becoming increasingly urbanized and distanced from these natural systems. The world’s climate system is an integral part of this complex of life-supporting processes, one of many large natural systems that are now coming under pressure from the increasing weight of human numbers and economic activities.

By inadvertently increasing the concentration of energy-trapping gases in the lower atmosphere, human actions have begun to amplify Earth’s natural greenhouse effect. The primary challenge facing the world community is to achieve sufficient reduction in greenhouse gas emissions so as to avoid dangerous interference in the climate system. National governments, via the UN Framework Convention on Climate Change (UNFCC), are committed in principle to seeking this outcome. In practice, it is proving difficult to find a politically acceptable course of action—often because of apprehensions about possible short-term economic consequences.

This volume seeks to describe the context and process of global climate change, its actual or likely impacts on health, and how human societies should respond, via both adaptation strategies to lessen impacts and collective action to reduce greenhouse gas emissions. As shown later, much of the resultant risk to human populations and the ecosystems upon which they depend comes from the projected extremely rapid rate of change in climatic conditions. Indeed, the prospect of such change has stimulated a great deal of new scientific research over the past decade, much of which is elucidating the complex ecological disturbances that can impact on human well-being and health—as in the following example.

The US Global Change Research Program (Alaska Regional Assessment Group) recently documented how the various effects of climate change on aquatic ecosystems can interact and ripple through trophic levels in unpredictable ways. For example, warming in the Arctic region has reduced the amount of sea ice, impairing survival rates for walrus and seal pups that spend part of their life cycle on the ice. With fewer seal pups, sea otters have become the alternative food source for whales. Sea otters feed on sea urchins, and with fewer sea otters sea urchin populations are expanding and consuming more of the kelp that provides breeding grounds for fish. Fewer fish exacerbate the declines in walrus and seal populations. Overall, there is less food available for the Yupik Eskimos of the Arctic who rely on all of these species.

Global climate change is thus a significant addition to the spectrum of environmental health hazards faced by humankind. The global scale makes for unfamiliarity—although most of its health impacts comprise increases (or decreases) in familiar effects of climatic variation on human biology and health. Traditional environmental health concerns long have been focused on toxicological or microbiological risks to health from local environmental exposures. However, in the early years of the twenty-first century, as the burgeoning human impact on the environment continues to alter the planet’s geological, biological and ecological systems, a range of larger-scale environmental hazards to human health has emerged. In addition to global climate change, these include: the health risks posed by stratospheric ozone depletion; loss of biodiversity; stresses on terrestrial and ocean food-producing systems; changes in hydrological systems and the supplies of freshwater; and the global dissemination of persistent organic pollutants.

Climate change and stratospheric ozone depletion are the best known of these various global environmental changes. Human societies, however, have had long experience of the vicissitudes of climate: climatic cycles have left great imprints and scars on the history of humankind. Civilisations such as those of ancient Egypt, Mesopotamia, the Mayans, the Vikings in Greenland and European populations during the four centuries of Little Ice Age, all have both benefited and suffered from nature’s great climatic cycles. Historical analyses also reveal widespread disasters, social disruption and disease outbreaks in response to the more acute, inter-annual, quasi-periodic ENSO (El Niño Southern Oscillation) cycle (1). The depletion of soil fertility and freshwater supplies, and the mismanagement of water catchment basins via excessive deforestation, also have contributed to the decline of various regional populations over the millennia.

[2]. Geomorphology, natural hazards, vulnerability and prevention of natural disasters in developing countries

The significance of the prevention of natural disasters is made evident by the commemoration of the International Decade for Natural Disaster Reduction (IDNDR). This paper focuses on the role of geomorphology in the prevention of natural disasters in developing countries, where their impact has devastating consequences. Concepts such as natural hazards, natural disasters and vulnerability have a broad range of definitions; however, the most significant elements are associated with the vulnerability concept. The latter is further explored and considered as a key factor in understanding the occurrence of natural disasters, and consequently, in developing and applying adequate strategies for prevention. Terms such as natural and human vulnerabilities are introduce and explained as target aspects to be taken into account in the reduction of vulnerability and for prevention and mitigation of natural disasters. The importance of the incorporation not only of geomorphological research, but also of geomorphologists in risk assessment and management programs in the poorest countries is emphasized.

[3]. Atmospheric and climatic hazards: Improved monitoring and prediction for disaster mitigation

The last few years have seen enormous damage and loss of life from climate and weather phenomena. The most damaging events have included the severe 1997/98 ElNiño (with its near-global impacts), HurricaneMitch, and floods in China in mid-1998. What have we learnt regarding the causes, variability, and predictability, of these phenomena? Can we predict the occurrence of these extreme events, and there by mitigate their damage? This paper reviews what we have learnt in the last decade or so regarding the predictability of these climate and weather extremes. The view starts with the largest (El Niño) scales, and works towards the scale of individual thunderstorms. It focuses on the practical outcomes of our improved knowledge with regard to decreasing the impact of natural disasters, rather than describing in detail the scientific knowledge underlying these outcomes. The paper concludes with a discussion of some of the factors that still restrict our ability to mitigate the deleterious effects of atmospheric and climatic hazards.

[4]. Geetha, A., and G. M. Nasira , have discussed regarding the rainfall prediction which was done using Decision Tree Algorithm. In their study , they have considered various weather conditions apart from only parameters, i.e cyclones, temperature etc at particular places. The prediction model starts with exploring dataset with calculating mean error values using normalization and then deploying the decision tree algorithm which showed pretty high accuracy around 80%.

[5] Thirumalai, Chandrasegar, et al. has published and discussed about predicting future rainfall using past year rainfall during crop seasons. These crop seasons are Rabi, Kharif and Zaid. The machine learning technique used in his study is Linear Regression method. This method uses one or more explanatory variables and to predict the case of independent variable. Here, rabi and kharif are the variables considered to predict an other variable seasonal rainfall.

[6] Steve Oberlin, et.al (2012) has proposed wide range of Machine Learning strategies for the procession of big data. He has included massive amounts of data sets. He has included the machine learning technique under Supervised Learning, Linear regression which again uses one or more exploratory variables in order to predict an independent variable.,he has also used preception and k means algorithm for discovering patterns in the current data or historic data. Perception, k- means which is under unsupervised learning are the few strategies used by him for uncovering the relationships and finding patterns in data. K means is a machine learning technique that is used for unlabelled data and puts them into clusters. These clusters are linked with a centroid or K point.

[7] Khan, Sharma, Mehrotra, Schepen, & Wang, 2015 have discussed regarding Bayesian joint probability model to predict post-processing rainfall when short stream rainfall prediction didn’t focus on floods. The dataset used was an Australian dataset.

[8] Kar, Kaveri, Neelima Thakur, and Prerika Sanghvi has presented their study which involves the ML technique fuzzy logic. Fuzzy logic is defined as many-valued variables considered between 0 and 1. This is more efficient considering Boolean logic. It delivers multiple logic variables using the same variable. Here they have taken only Temperature initially as an atmospheric parameter, but considering with respect to fuzzy logic, temperature alone couldn’t make it to prediction, hence they have also considered, humidity as another parameter.

[9] Chen, Binghong, et al. have presented their study and practical approach by using non-linear algorithms in the field of Machine Learning, these include gradient boosting decision tree model, and deep neural networks in order to predict rainfall. It’s errors are calculated by using F1 scores and RMSE i.e Root mean square method. It was built on Alibaba cloud. The data was collected from wide range of sites which were different.

[10] Considering the usage of machine learning techniques, there are wide range of algorithms that are available, for example, LASSO regression is one form of algorithm that is used in the process of prediction. Also, ANN – Artificial neural network approach can also be done for prediction. However, the approaches that are mentioned just now are considered to possess less percentage when it comes in terms of accuracy. To reduce these accuracies, comparison of numerous techniques must be done in order to find the machine learning technique that possesses high accuracy.

[11] Parmar, Aakash, Kinjal Mistree, and Mithila Sompura have discussed the regarding the usage of various machine learning techniques to predict rainfall using weather conditions. These techniques are Neural networks, ARIMA model, Back Propagation Neural network as well as SVM- Support vector Machines. The Neural networks come under deep learning which is again a subset of machine learning. These networks have hidden layers in it which determine patterns associated. Most commonly used neural networks are conventional neural networks. Back propagation algorithm is the one that is used to calculate derivates. By the name itself it suggests that, the weights are updated from output to input. ARIMA model is defined as Auto regressive integrated moving average and is used to predict future values using past values of time series.

5. Exploratory Data Analysis: the Best way to Start a Data Science Project

Exploratory Data Analysis is a set of techniques that were developed by Tukey, John Wilder in 1970. The philosophy behind this approach was to examine the data before building a model. John Tukey encouraged statisticians to explore the data, and possibly formulate hypotheses that could lead to new data collection and experiments. Today Data scientists and analysts spend most of their time in Data Wrangling and Exploratory Data Analysis also known as EDA. But what is this EDA and why it is so important? This article explains what is EDA and how to apply EDA techniques to a dataset.

**OBJECTIVES AND OVERVIEW OF THE REPORT :**

* The predictive model is used to prediction of the precipitation. The first step is converting data in to the correct format to conduct experiments then make a good analysis of data and observe variation in the patterns of rainfall.
* We predict the rainfall by separating the dataset into training set and testing set then we apply different machine learning approaches ( Bagging classifier and Xgboost ) and compare and draw analysis over various approaches used. With the help of numerous approaches we attempt to minimize the error.
* This study provides the practical implementation of Machine Learning techniques that are used to possess high accuracy. Commencing from the scratch, Machine Learning is the subset of Artificial intelligence. Machine Learning is generally defined as the computer systems that identify or extract inferences or let’s say patterns from the input data without any human interaction following by the training of the system with various Algorithms. ML is ruling the world ever since the concept of Artificial intelligence is introduced.
* **Python (version 3.8)** is being used in this study as it a general-purpose language and a high-level language which can be easily handled as compared to other high-level programming languages. Python uses an object-oriented approach which uses logical code and it is dynamically programmed.
* **Anaconda** is the distribution of programming languages. One among the distribution is Python language. It is used for scientific computing such as in terms of Data Science. Anaconda was initialized about 9 years ago in 2012. Also there is conda, which manages all the packages that have to installed under Anaconda. There’s also Anaconda Navigator which is a GUI application and it’s main purpose is to launch applications which do not require Command-line prompt to launch an application. This Navigator has wide range of applications, where in , for this study we are going to consider, Jupyter Notebook.
* **Jupyter Notebook** is an open-source software which runs programs in the language of Python. This notebook allows editing and runs rich documents. Wide range of libraries are imported in order to suit the considered algorithm.
* The errors in the values of dataset are handled in such a way that the errors are reduced. Also, quality of dataset is very much prominent. This quality can define the whole process of prediction. The quality of dataset that is used here is the influences the whole process. This process is also known as decomposition technique. This technique is the one that connects with the quality of dataset being used. Then comes the way of using ensemble learning method which defines the process of standard way of predicting future result with respect to the current data or historical data. The past data in the dataset plays a major role as it includes numerous amounts of parameters which boosts the techniques or algorithms that are used in the field of prediction.

**SYSTEM REQUIREMENTS :**

* **Hardware Requirements :**
* **System** : MINIMUM i3.
* **Hard Disk** : 40 GB.
* **Ram** : 4 GB.
* **Software Requirements :**
* **Operating System:**  Windows 8
* **Coding Language**: Python 3.7
* Included with Jupyter notebook..
* Anaconda Navigator (python 3.8)
* **Atmospheric Parameters** like mintemp, maxtemp, humidity, etc from the Meteorological Department

**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 and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**SYSTEM DESIGN :**

**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.

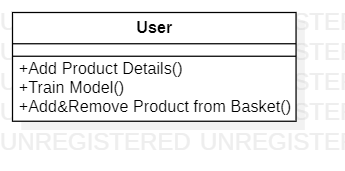
**USE CASE DIAGRAM:**

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.



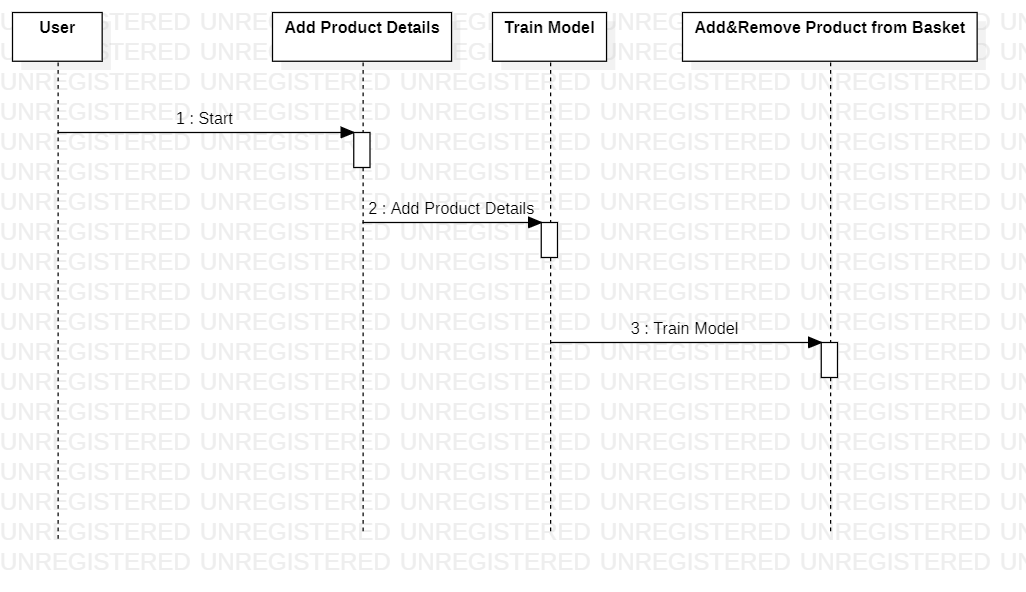
**CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

****

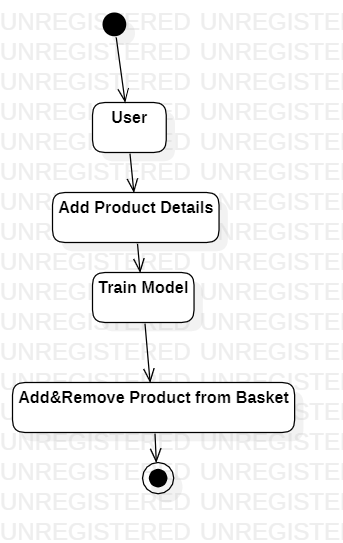
**SEQUENCE DIAGRAM:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

****

**ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**IMPLEMENTATION:**

**MODULES:**

1.Add Product Details

To build project I used some sample products image to train product identification models

2.Train Model

In this Module screen train model generated with 100% accuracy and now show product to web cam.

3. Add/Remove Product from basket

To allow application to identify product image and then show in text area and if we again show same product then application will remove from text area

**SOFTWARE ENVIRONMENT :**

# 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.

### **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.

**The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.**

#### 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.

### **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.

**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."

**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.

**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.

## 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.

## 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 −

1. **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.
2. **Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.
3. **Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.
4. **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.
5. **Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.
6. **Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.
7. **Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

## 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

# 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**

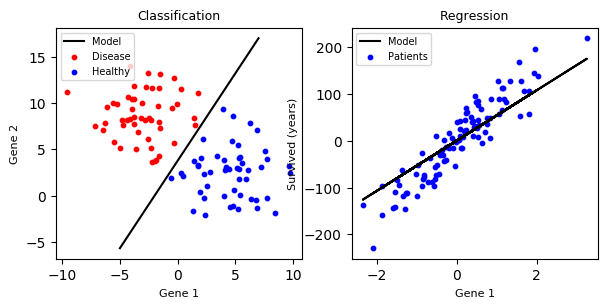
Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It’s best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

#### (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. This is one of a type in Machine Learning. This type of learning is one such learning which uses Labelled data. This labelled data is the input data which will be trained with a particular algorithm. As the name itself suggests that the learning is under a Supervisor. This supervisor is the labelled data. Based on the labelled data, the output type can be predicted. There are two categories in supervised learning.
* **Classification** : This one type of category where in the output is in the form of classified type. For example, considering this project , whether it will rain tomorrow or not, the output will be in form of either ‘yes’ or ‘no’. Classification is widely used machine learning category. It has some amount of algorithms which have high accuracy and computational speed.
* **Regression** : It is an other type of category where the output is in the form of a real value. Both classification and regression are used in Prediction.



* **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.

### **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.

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

#### 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.

#### 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.

#### 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.

#### 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.

**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 nicode.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

**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.

**Modules/ Libraries 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. For machine learning or data analysis, where ever data is used , pandas comes into play. With pandas one can merge data , reshape data, join data along with analising data.

* **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. It is largely written in python, and uses numpy for high performance calculations and array operations. Scikit learn has been very popular library started with machine learning. Scikit was originally known as scikit.learn which was invented in the google summer of code by David Cournapeau. The part of the name Scikit has come from SciPy Toolkit. Since then it has become a well documented and popular python machine learning library. It provides well reliable code for machine learning algorithms that are under both supervised and unsupervised learning. It is used for classification, regression and clustering problems as well. It’s an amazing way to known how these different models do and gain some intuition out of them when various parameters are been given and to perform.

**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.

**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**

**SYSTEM TEST :**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### **TYPES OF TESTS**

* **Unit testing :**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

* **Integration testing :**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

* **Functional test :**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

**Valid Input** : identified classes of valid input must be accepted.

**Invalid Input** : identified classes of invalid input must be rejected.

**Functions** : identified functions must be exercised.

**Output**  : identified classes of application outputs must be exercised.

**Systems/Procedures** : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

* **System Test :**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

* **White Box Testing :**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

* **Black Box Testing :**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

* **Unit Testing :**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

* **Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

1. **Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

1. **Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# Integration Testing :

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

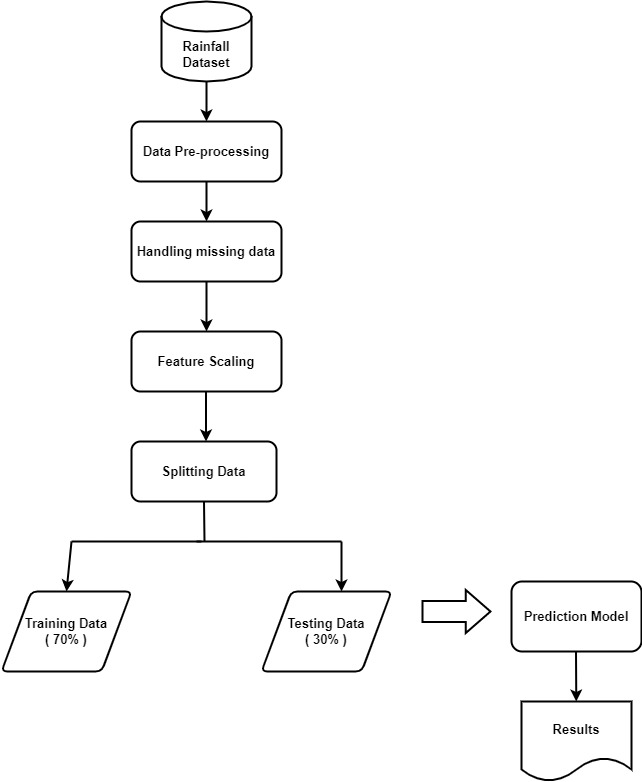
* **Acceptance Testing :**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**METHODOLOGY :**

Our prediction model goes through various phases, and these are depicted in the following architecture. Input data is taken from the dataset which also has target variable. Target variable is taken in the form of binary classification since output is not accepted in the form of alphabets.



**Fig : System Architecture**

The overall architecture include four major components: Data Exploration and Analysis, Data Pre-processing, Model Implementation, and Model Evaluation, as shown in Fig.

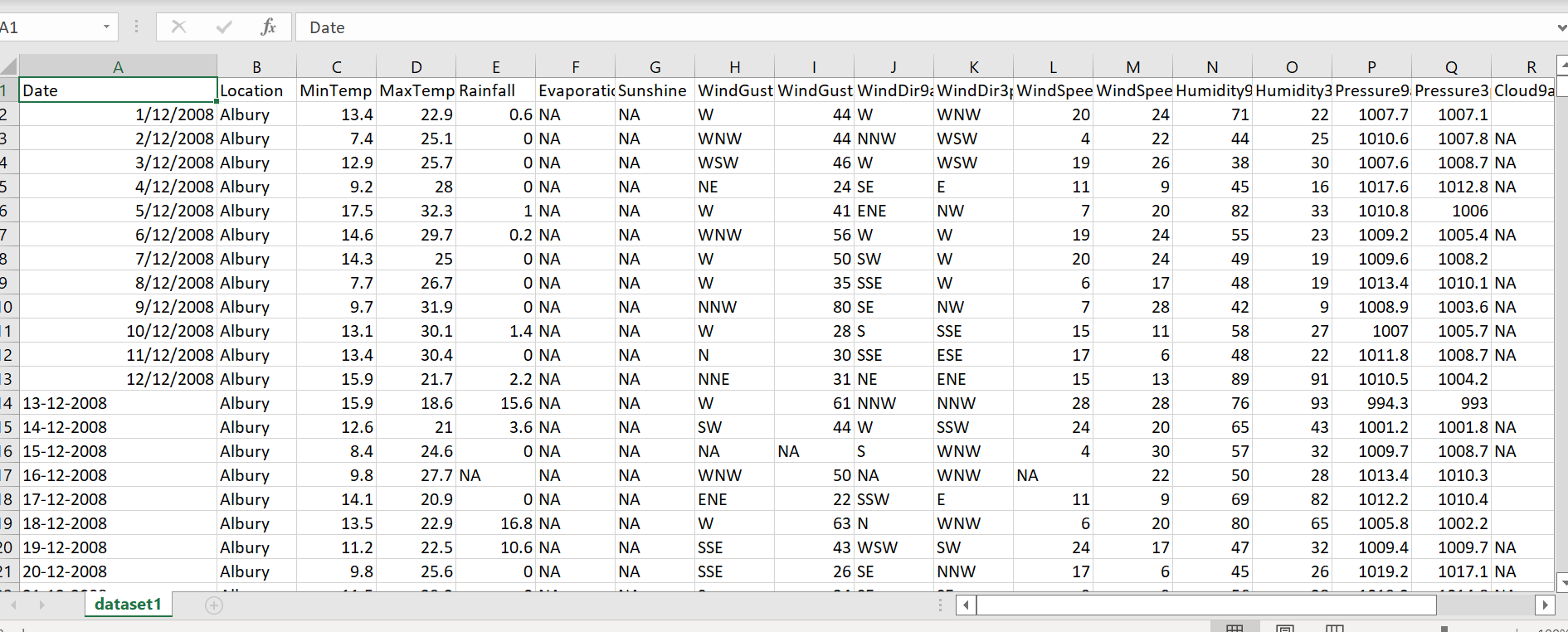


***Fig :Over All Architecture***

**Step 1 : Data Exploration and Analysis :**

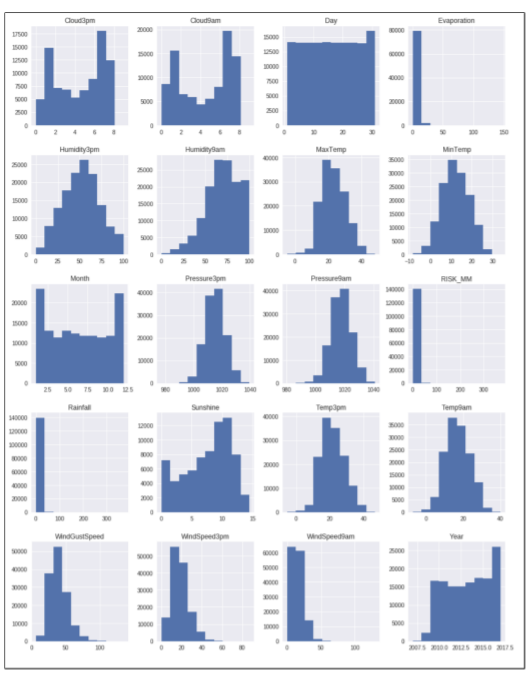
Exploratory Data Analysis is valuable to machine learning problems since it allows to get closer to the certainty that the future results will be valid, correctly interpreted, and applicable to the desired business contexts [4]. Such level of certainty can be achieved only after raw data is validated and checked for anomalies, ensuring that the data set was collected without errors. EDA also helps to find insights that were not evident or worth investigating to business stakeholders and researchers.

Here, we’re going to consider the dataset in a CSV file which means comma separated values. The dataset contains various input parameters which are atmospheric parameters from the meteorological department. These are collected from Kaggle website.

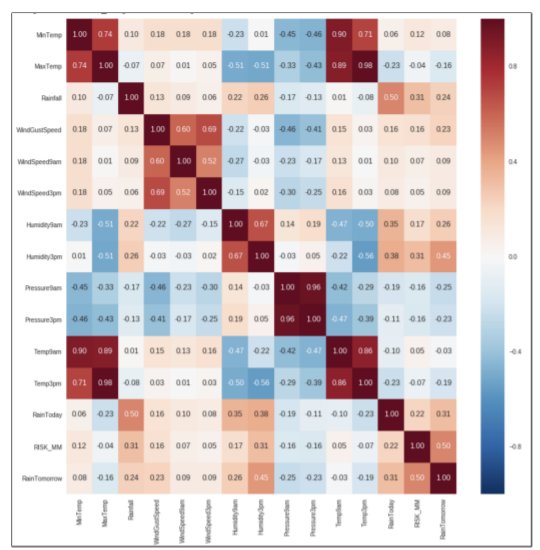


**Fig : Dataset in csv file**

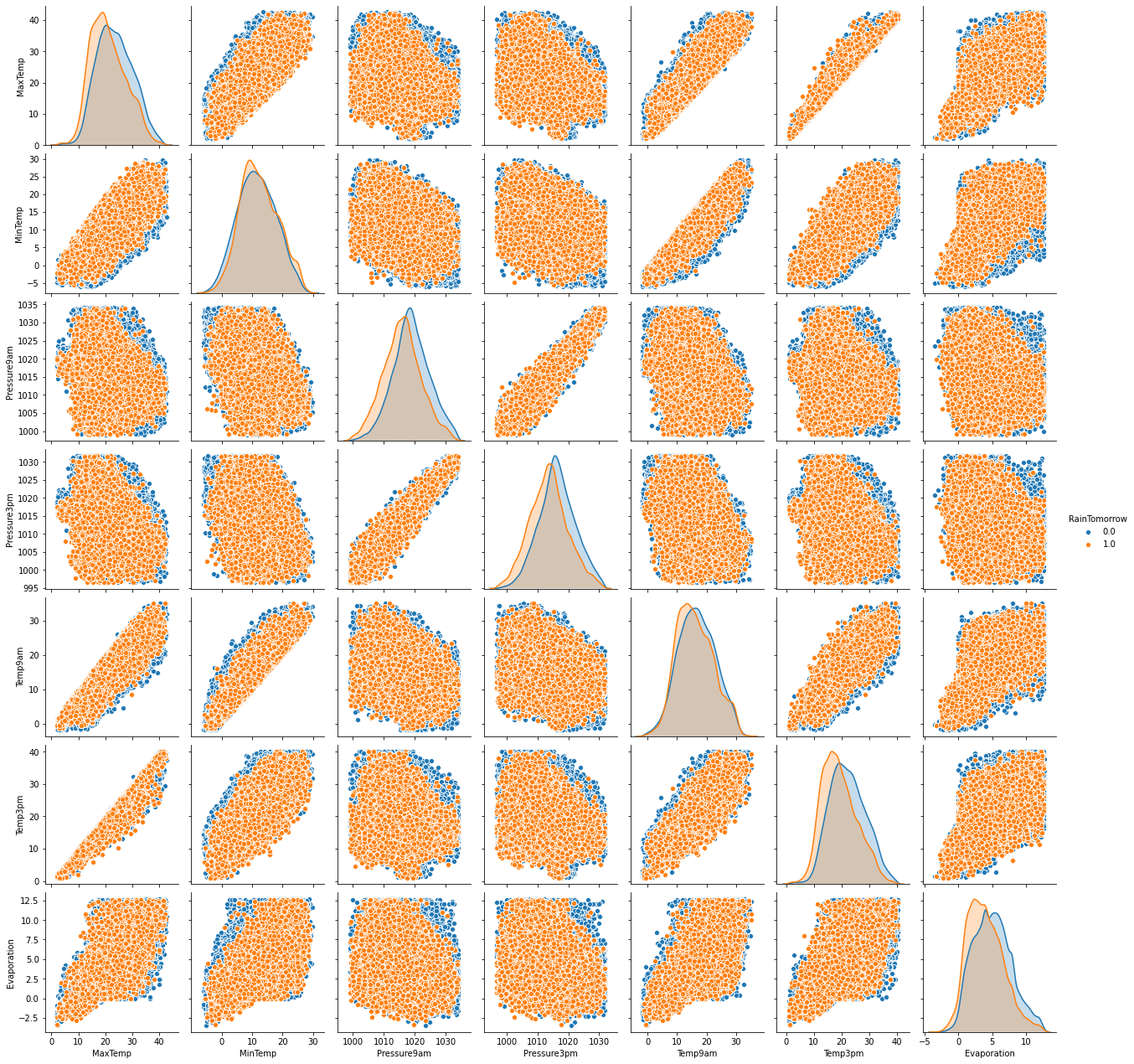
We performed EDA using two methods - Univariate Visualization which provides summary statistics for each field in the raw data set (figure 2) and Pair-wise Correlation Matrix which is performed to understand interactions between different fields in the data set.



**Fig : Univariate Distribution**



**Fig : Heat map**



**Step 2 : Data preprocessing :**

* Data preprocessing is a data mining technique which transforms raw data into understandable format.
* **Dealing invalid data or Missing values** : The dataset may contain null values. To deal or impute the missing values, we will group and replace the null values by their respective mean values.

**from** **sklearn.impute** **import** SimpleImputer – is used to impute missing va -lues

**fit\_transform** – first fits the data and transforms it.

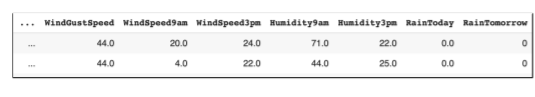
* **Normalizing data** : Converting the data into machine readable format.

***class* sklearn.preprocessing.LabelEncoder –** is used for normalizing values. Label encoder is also used for non-numerical values.

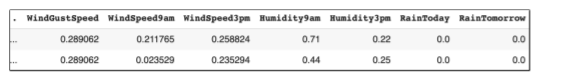
* **Feature scaling**: : Our data set contains features with highly varying magnitudes and range. To suppress this effect, we need to bring all features to the same level of magnitudes. This can be achieved by scaling.

**sc = StandardScaler()** – it’s function is to scale the input values by removi -ng it’s mean to unit variance or scaling values to a particular range.

* **Feature Selection:** Feature Selection is the process where you automatically or manually select those features which contribute most to our prediction variable or output. Having irrelevant features in data can decrease the accuracy of the models and make the model learn based on irrelevant features.



**Fig : Before scaling**

 **Fig : After scaling**

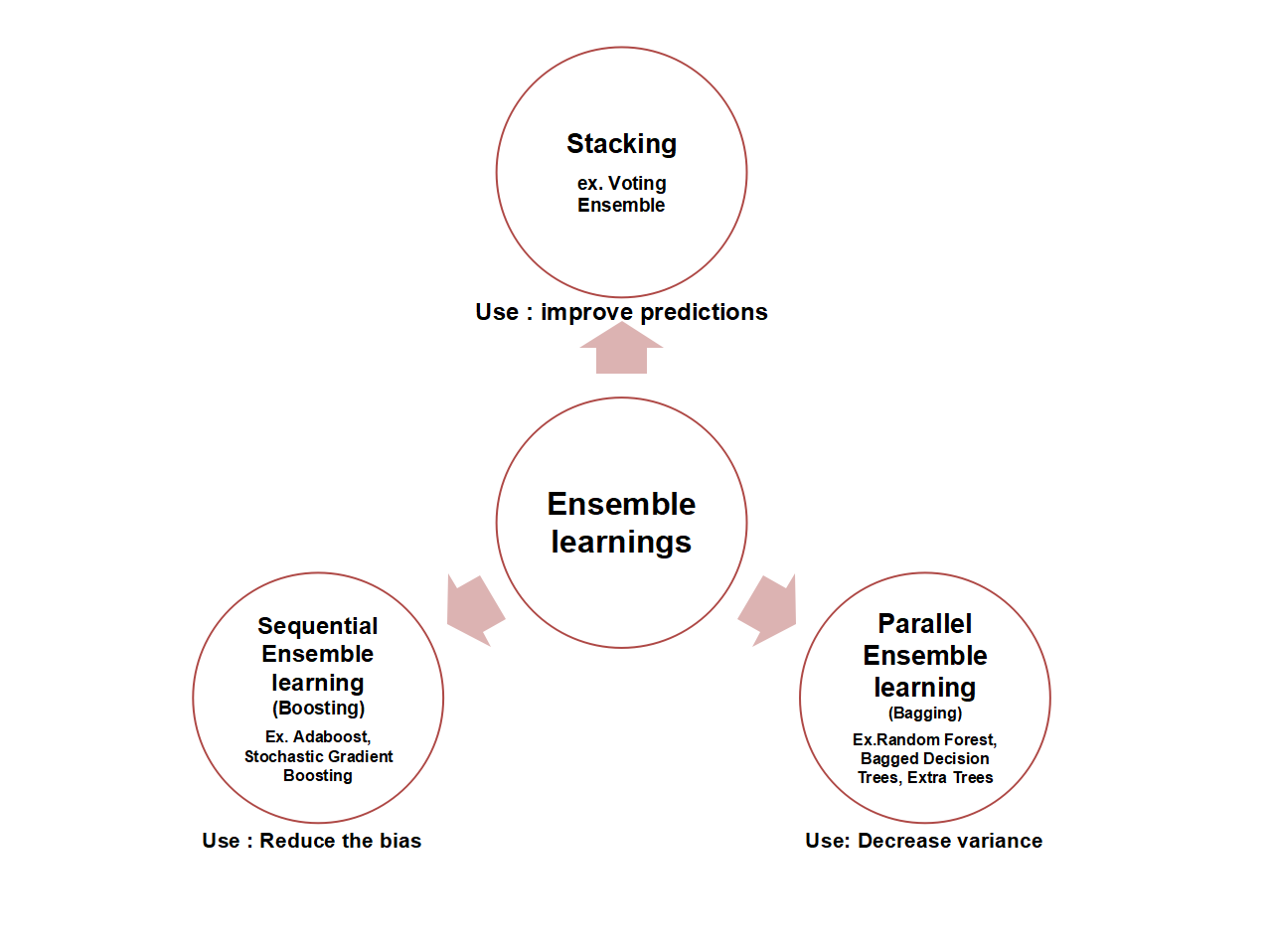
**Step 3 : Splitting data into training and testing phases:**

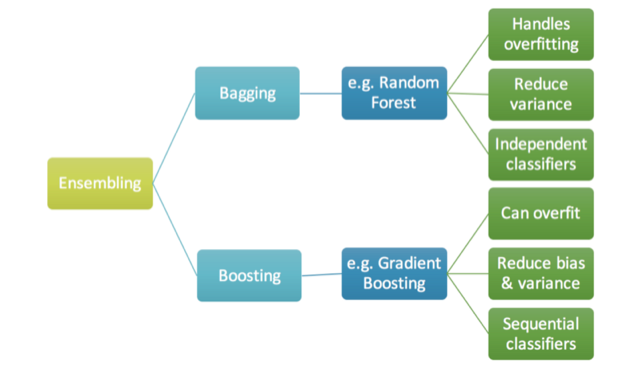
* Using the same dataset for both training and testing chances for miscalculations, thus increases the chances of inaccurate predictions.
* The ideal split is said to be 80:20 for training and testing. You may need to adjust it depending on the size of the dataset and parameter complexity.
* **train\_test\_split :** It is used for splitting data into random training and testing sets.

**Step 4 : Modelling and evaluation :**

**Machine learning algorithms used in the EXISTING SYSTEM as well as in the PROPOSED SYSTEM:**

Here, we are going to consider **Ensemble Learning** under machine learning. Ensemble Learning is the process of combining multiple models and obtaining prediction considering the overall performance of these models. Suppose if we want to buy a car, we just won’t land up to the conclusion by just visiting one particular dealer. We generally perform research regarding car models and comparing certain car models and it’s features, we finally land up to the conclusion. In a similar way, Ensemble learning concept works. Various algorithms are considered under ensemble learning. We train the input data into these multiple models and prediction of output is done. The two main advantages of ensemble learning are it ensures reliability of predictions and it also ensures robustness of the model or let’s say stability of the model.





The above figures depict two techniques under Ensembling Learning. These are **Bagging** and **Boosting.** Bagging is also known as **Bootstrap aggregator.** This process of bootstrapping means taking random samples of dataset with replacement. Bootstrapping also avoids the problem of overfitting . It performs well with the test data by reducing the variance when training data is given to multiple base learners. For classification, there is a term called hard voting which means that it takes the majority vote from multiple models. Whereas for regression there is soft voting which calculates the average probability of these multiple models. Bagging is stable against noice.The examples of Bagging include Random Forest which can handle overfitting of data, and can reduce variance. Bagging generally involves Parallel classifying. Boosting is generally used to reduce bias and variance Boosting has gradient boosting, Xg boost as examples and it may overfit the data and can reduce both bias and variance. Boosting generally contains sequential classifying. There is mathematics involved behind these algorithms. The geometrical patterns containing in these models will be discussed further.

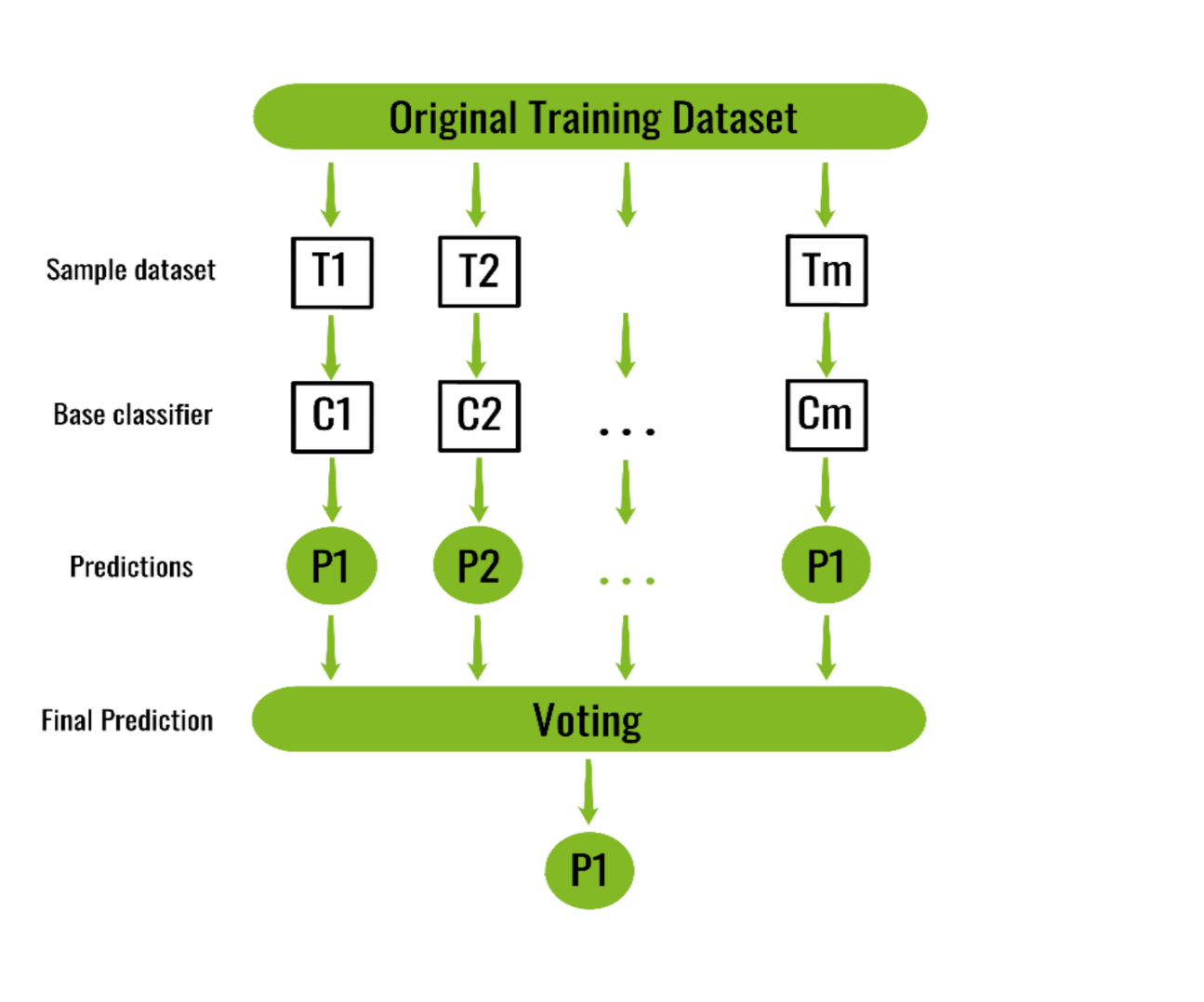
* **Random forest :** Random forest classifier involves various decision trees, where a decision tree consists of root node, leaf node and decision nodes. These decision trees are building blocks of random forest. Based on the training data fed to the decision tress, each produce different output and the one that is highest is ranked first i.e majority in voting of outputs is considered in the algorithm of random forest. Random forest is highly accurate than decision tress, infact it boosts up the decision tree algorithm in an effective way. In Random forest classifier, we’re firstly going to consider a particular dataset. And then we use decision trees as multiple models. From the original dataset which consists of several rows and columns, we consider some sample of input data from rows as well as columns. This process is known as Row sampling with replacement and Feature /column sampling with replacement. Here with respect to decision trees, let’s say there are DT1, DT2…..DTn which are several decision trees. The sample data which is sent to these decision trees is always less than the original dataset . In other words, the rows and column data which is considered for combing with decision trees is always less than the row and column data which original dataset contains. Then each and every decision tree gets trained with this sample input data. Then each decision tree predicts some output value. Here, majority of voting is taken at the end. Every decision tree delivers some output value, and final prediction is taken for the one which has highest no of votes. Basically, decision trees has two main properties that are : It has low bias and high variance. Low bias comes in point when we’re creating a decision tree and exploring it to it’s complete depth. Low bias allows the whole decision tree to get properly trained with the training dataset. This results in low error of training data. Meanwhile, high variance means that the test data may contain high errors. But when we combine multiple models of these decision trees and obtain majority voting, this high variance will be converted to low variance. When we give particular records as sample input data to the decision tree, the tree becomes an expert in considering these particular records, since each decision tree may contain row and feature sample data which can be repeated in other trees as well. That is why it is termed as row/column sampling with replacement. Even if there are change in rows or columns, these changes won’t affect the accuracy or the output prediction value since the input records will split to the decision trees. Hence, whenever there is a change in data or whenever there is a change in these records of input data, the high variance will still be converted to low variance. There can be high chances of possessing low amount of error in the data. And the random forest classifier that sklearn uses , will consider the majority voting of the decision trees. Whereas, random forest regressor will consider the average or mean of the output values delivered by each and every decision tree. The name itself implies that forest means considering several amount of decision trees, and randomly the input data is being selected. Random forest is one such algorithm which is under supervised learning and combines these multiple models to improve the overall performance. It also has various advantages which include less training time as compared to other algorithms. And can even take large amount of datasets. The advantages of random forest include high flexibility and is compatible for both classification and regression tasks. Works well with both categorial and continuous variables. Normalization of data is not required as random forest uses rule based approach. There are many drawbacks too that are associated with this algorithm. Since it uses multiple decision trees, the computational power is higher. And it’s training time is comparatively higher as it has to produce outputs based on consideration several decision trees. An other drawback that is associated with random forest is that it suffers interpretability because of ensemble decision trees and fails to depict significance of each variable. Random Forest applications are it is used in banking sector and healthcare sector as well.



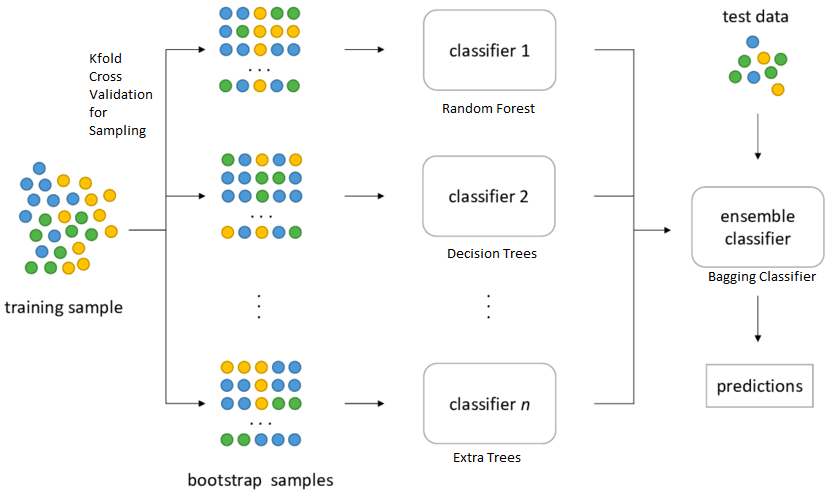
Here, we’re considering random forest classifier and importing it as well as calculating it’s accuracy.

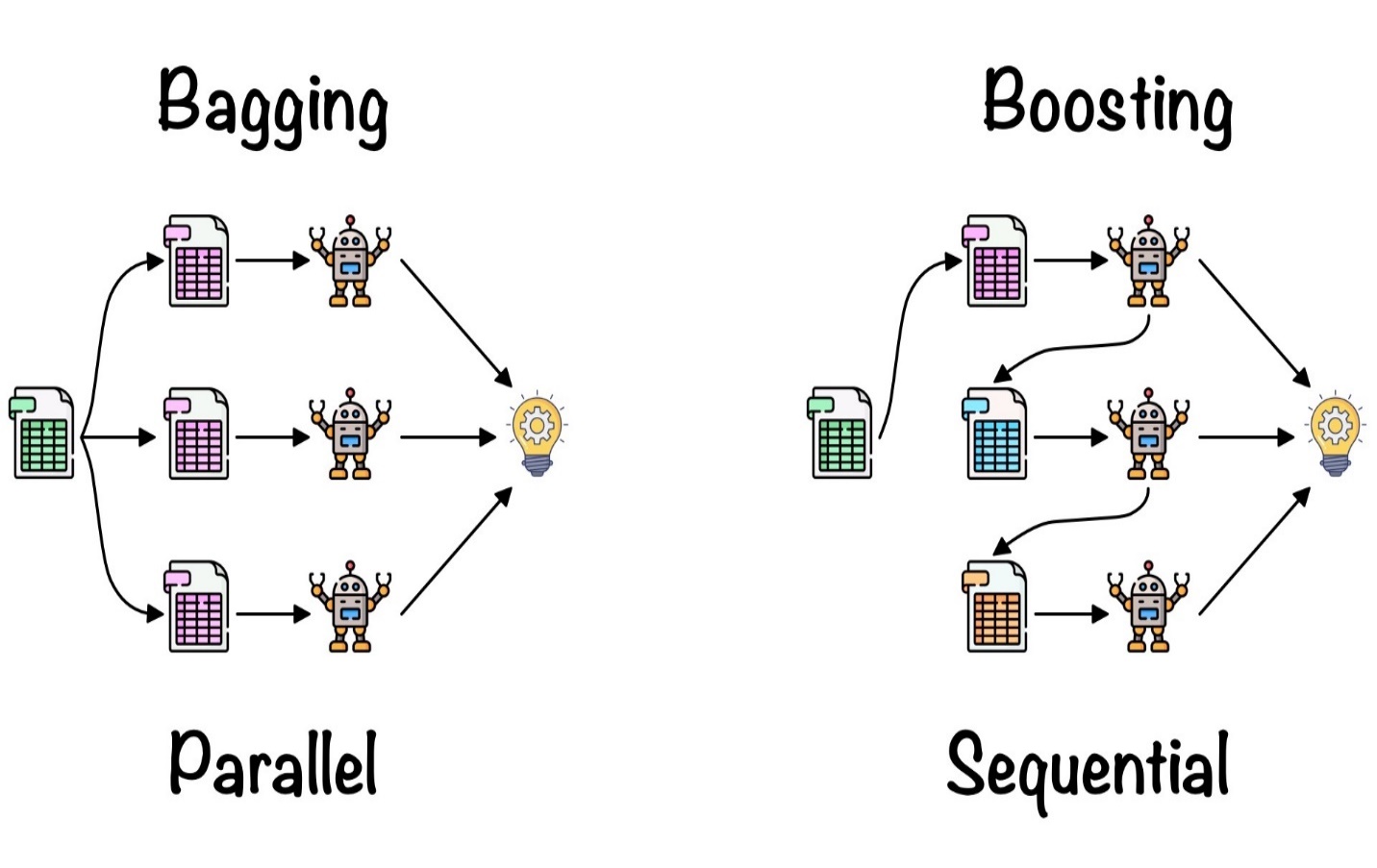
classifier = RandomForestClassifier(n\_estimators=100,random\_state=0), then we’re trainng into two different sets i.e X and Y, fitting the data into X\_train and Y\_train and then applyiong inverse transform function is to scale back the data into it’s original representation. Then testing phase is executed and accuracy score of random forest classifier is calculated.

* **Bagging classfier** : A Bagging classifier is regarded as an ensemble meta-estimator which fits each and every base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. This is one such meta-estimator which can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble method . It prevents overfitting of data. Firstly, N sample instances of data are given and can be replaced through which each base classifier will get trained in parallel. Suppose there is dataset which consists of several rows and coloumns, also there are several multiple models let’s say C1, C2…..Cm, here the dataset is firstly combined to C1, only random samples of this input data is being combined to the model C1. For the next model/ base classifier C2, we consider some other set of samples, and combine them to the model. This process is known as Row sampling with replacement. Here the sample input present in a particular model or let’s say base learners, are less than the original dataset. Here considering the below figure, we can observe that the sample input data which is taken has T1, T2,….Tm. Every base classifier is being trained with their respective sample dataset. Now considering the test data, when the test data is being sent to every base classifier, there are predictions that these base classifiers deliver. In the below figure these predictions are P1,P2…..P1. Further more, the predictions which have higher no of voting becomes the final prediction value. P1 is the final prediction in the below figure as it has the highest majority of voting. The process of row sampling with replacement is termed from Bootstrap. Infact, this process is known as Bootstrap. The final prediction value process is known as Aggregator. This aggregator delivers the final output value. Bagging is one such method in which the base learners are trained parallelly. Bagging classifier is easy to implement as it is available in sklearn and one more advantage that bagging possesses is the reduction in variance especially in regard with high dimensional data which can promote overfitting. Bagging is computationally expensive and has major applications involved. Bagging is mainly used for predictions. It is used in health care for making predictions on medical data. And it is also being used in the IT environment. Used in finance as well concerning with deep learning models. In the IT sector, bagging is being used in network intrusion detection systems and many more. One main advantage of this bagging classifier is it reduces the variance by combing multiple models. It is used in banking sector as well as in high- dimensional data. It is also applied in land cover mapping. Bagging is completely a data specific algorithm. Considering the hyperparameters, Scikit uses two main paramaters i.e bagging classifier and regressor. Since our requirement is classification we use bagging classifier. Base\_estimator which defines the random subsets. There is n\_estimator which defines the no of base estimators. Random\_state and n\_jobs. It has robust architecture.

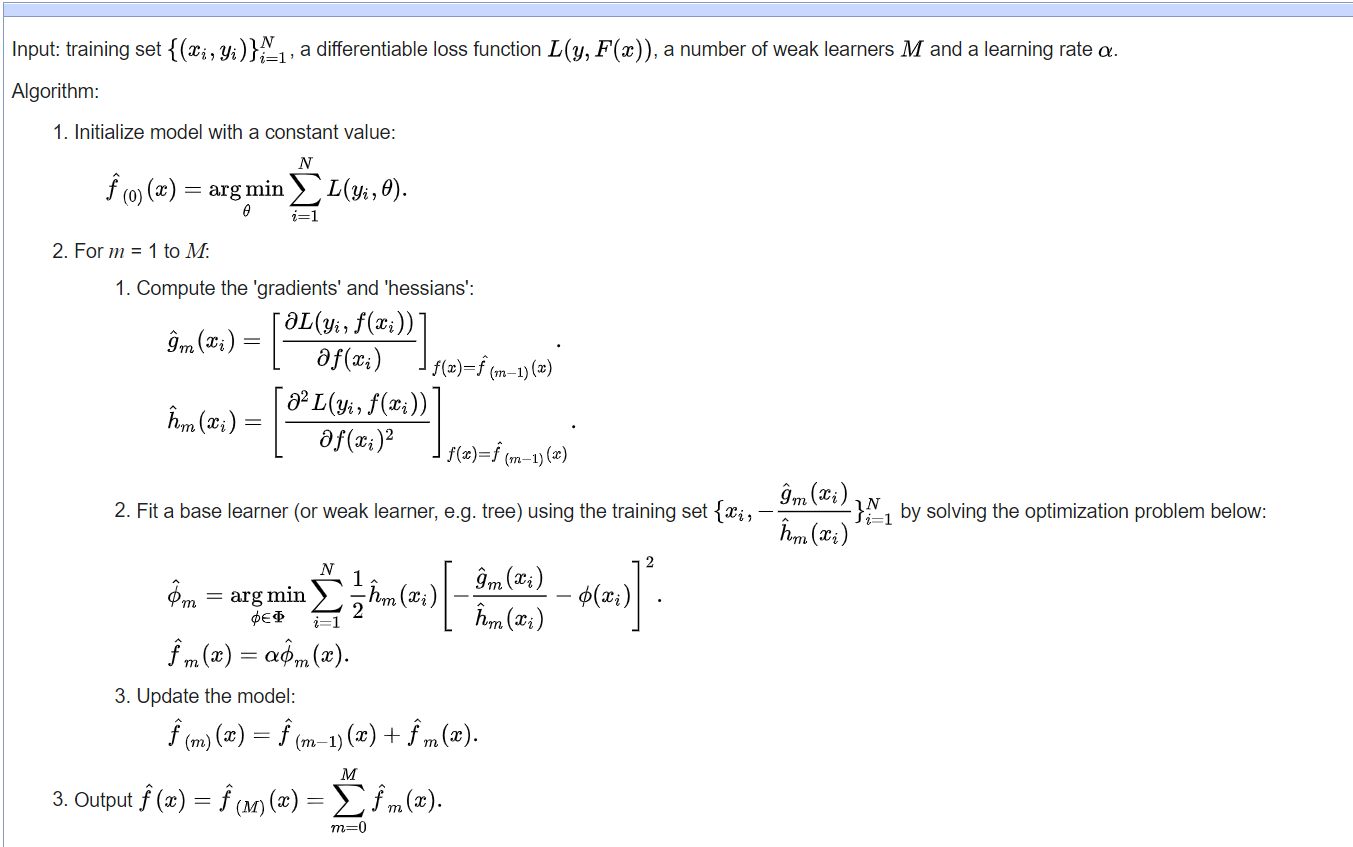


**from** **sklearn.ensemble** **import** BaggingClassifier is considered and prediction of rain as well as calculating it’s accuracy score are also being done.





* **XG Boost classifier**  : It is one such classifier which is highly efficient and portable. It is being implemented by the library xgboost and makes predictions considering some amount of models, known as base learners in order to make a predicition. It uses trees which are called CART ( classification and regression trees ). Xgboost will optimize training for Gradient boosting. Initially independent features are considered and a target variable which is the output. The target variable will be in form 0 or 1 since it’s a binary classification problem. Now extracting the average of this target value, base model is considered and this probability value is taken. In the similar way which gradient boosting works, we subtract the target variable with the probability value which is the average. Now comes the residual values for every feature. The next step is constructing decision trees. To initialize this construction, we have data points as residuals. One particular feature becomes a root node, and the residual values are categorized based on the root node. In xgboost classifier the decision tree leaf nodes must range between 0 to 1. The second step is calculating similarity weights. These similarity weights are calculated for every node separately. The third requirement is calculating information gain. The similarity weights are calculated for every node including the root node of a particular decision tree. Xg boost has several advantages, which has reguralisation. And implements parallel processing much faster than gradient boosting. It also has high flexibility. It has built-in cross validation and Xgboost also promotes tree pruning. It is used in distributed computing for training very large models using a cluster of machines. The implementation of the algorithm was developed for efficiency of compute time and memory resources. A design goal was to make the best use of available resources to train the model. Few of the key algorithm features include: Sparse aware through which implementation is done by handling missing values. Block structure to support parallelization of tree construction. The third feature is continued training so that we can boost further an already fitted model on new data. The fact that Xgboost is high at execution speed is true. Xgboost is faster and works effienctly. And the model performance is better. XGBoost dominates structured or tabular datasets on classification and regression predictive modeling problems. We can use Xg boost as a framework and as well as built-in algorithm. Xgboost is an improved version of gradient boosting. It’s features include shrinkage and column sub-sampling. Xgboost is faster as compared to other algorithms because of it’s parallel and distributed computing. The main objective of Xgboost is to push the extreme computational limits of machines to provide a scalable , portable, and accurate library.

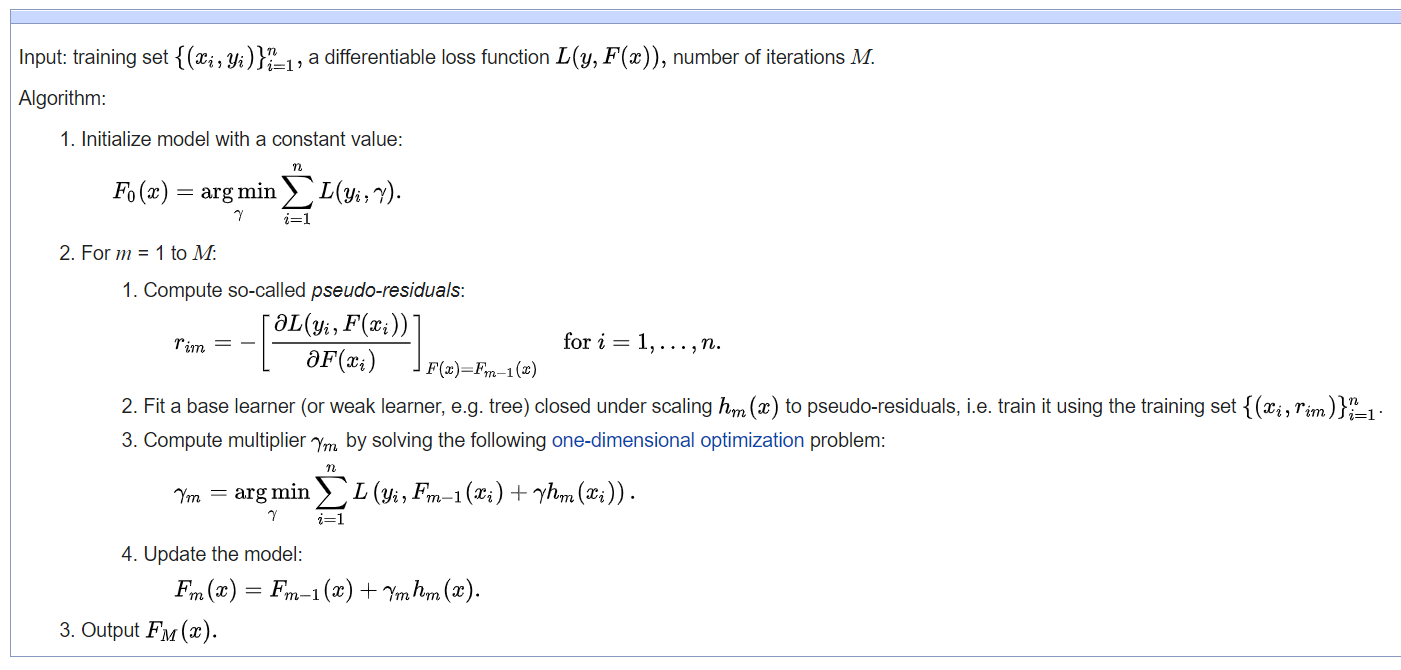




* **Gradient boosting**  : It is one of machine learning techniques under supervised learning which is used for predicting models. This considers weak models such as decision trees. Here , learning happens by optimizing the loss function. The trees of gradient boosting are comparatively larger as compared to adaboost . Gradient boosting also requires additive model. Initially the average of the target parameter is calculated. Then actual and prediction values are considered when there is generation of residuals and then updating the values of prediction. Firstly to begin with gradient boosting, we consider the average of the target variable. Then we consider base model value as this average value. As discussed earlier, gradient boosting also uses decision trees as base models. In the next step, we compute residuals. Residuals means errors, we compute these residuals which are also known as Pseudo residuals. These errors are being computed by loss function. This loss function means usage of mean squared, root mean squared, etc. But since we are considering a classification algorithm i.e gradient boosting, there are many loss functions under classification algorithms. They are hinge loss, log loss and many more. Then we subtract the actual target variable from the average of the target variables. The subtracted values become the new residual values. After the step of base model, we try to construct decision trees sequentially. Boosting is one type of ensemble technique and it works sequentially. Hence, we construct decision trees sequentially. For constructing decision trees, we take input parameters excluding target variable but the output must be considered as new residual values. Now after base model, with independent features we train the data by constructing decision trees. And output is residual value. The very next step we again train the decision tree by taking these independent features once again. Here, we’re getting the output as new residual value which is different from the old residual value generated during first time. Next we add the original average value and the very new residual value which is generated with the decision tree. And comparing this new output with the target variable, we check the accuracy that is being delivered by the algorithm. If the output is greater than the target variable then it leads to overfitting of data. Ultimately what we want is low bias and low variance. But this overfitting of data can lead to high variance. Hence when adding average value with the new residual value, we also consider learning rate with the residual value. This learning rate ranges between 0 to 1. We come to know that by taking several decision trees the residual values keep decreasing till it matches with the original target variable value. When the test data is being taken it has to pass all these decision trees in order to get the correct prediction.

The Pseudo code for the algorithm of gradient boosting is depicted in the below figure. In the below figure we’re considering input as ( xi,yi ) which are the independent and dependent features and also input involves loss function. This loss function must be differentiable i.e we should make derivative out of it. The third requirement in the input must be no. of trees that we want to give. The very first step is initializing the base model with a constant value. As discussed earlier this constant value is the average value. Then computing the loss function we subtract the target values by performing first order derivative. F(x)^2 is being calculated finally and this becomes the minimal value and fits into the model. Then we start implementing the steps mentioned in the pseudo code. This is how gradient boosting algorithm works. Finally, gradient boosting has three main components. These are loss function, weak learner and an additive model.







**Python code implementation with respect to 4 classifiers :**

**import** **numpy** **as** **np**

**import** **pandas** **as** **pd**

**from** **sklearn.impute** **import** SimpleImputer

**from** **sklearn.preprocessing** **import** LabelEncoder

**from** **sklearn.preprocessing** **import** StandardScaler

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

**from** **sklearn.ensemble** **import** RandomForestClassifier

**from** **sklearn.metrics** **import** accuracy\_score

dataset = pd.read\_csv ('Dataset/weatherAUS.csv',nrows=4000)

X = dataset.iloc[:,[1,2,3,4,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21]].values

Y = dataset.iloc[:,-1].values

print(X)

[['Albury' 13.4 22.9 ... 16.9 21.8 'No']

['Albury' 7.4 25.1 ... 17.2 24.3 'No']

['Albury' 12.9 25.7 ... 21.0 23.2 'No']

...

['BadgerysCreek' 10.0 22.4 ... 17.7 20.7 'No']

['BadgerysCreek' 4.6 28.7 ... 14.5 25.3 'No']

['BadgerysCreek' 7.8 27.8 ... 18.7 27.4 'No']]

print(Y)

['No' 'No' 'No' ... 'No' 'No' 'No']

Y = Y.reshape(-1,1)

*#Dealing with invalid Data*

imputer = SimpleImputer(missing\_values=np.nan,strategy='most\_frequent')

X = imputer.fit\_transform(X)

Y = imputer.fit\_transform(Y)

print(X)

[['Albury' 13.4 22.9 ... 16.9 21.8 'No']

['Albury' 7.4 25.1 ... 17.2 24.3 'No']

['Albury' 12.9 25.7 ... 21.0 23.2 'No']

...

['BadgerysCreek' 10.0 22.4 ... 17.7 20.7 'No']

['BadgerysCreek' 4.6 28.7 ... 14.5 25.3 'No']

['BadgerysCreek' 7.8 27.8 ... 18.7 27.4 'No']]

*#Encoding Dataset*

le1 = LabelEncoder()

X[:,0] = le1.fit\_transform(X[:,0])

le2 = LabelEncoder()

X[:,4] = le2.fit\_transform(X[:,4])

le3 = LabelEncoder()

X[:,6] = le3.fit\_transform(X[:,6])

le4 = LabelEncoder()

X[:,7] = le4.fit\_transform(X[:,7])

le5 = LabelEncoder()

X[:,-1] = le5.fit\_transform(X[:,-1])

le6 = LabelEncoder()

Y[:,-1] = le6.fit\_transform(Y[:,-1])

print(X)

[[0 13.4 22.9 ... 16.9 21.8 0]

[0 7.4 25.1 ... 17.2 24.3 0]

[0 12.9 25.7 ... 21.0 23.2 0]

...

[1 10.0 22.4 ... 17.7 20.7 0]

[1 4.6 28.7 ... 14.5 25.3 0]

[1 7.8 27.8 ... 18.7 27.4 0]]

print(Y)

[[0]

[0]

[0]

...

[0]

[0]

[0]]

Y = np.array(Y,dtype=float)

print(Y)

[[0.]

[0.]

[0.]

...

[0.]

[0.]

[0.]]

*#Feature Scaling*

sc = StandardScaler()

X = sc.fit\_transform(X)

print(X)

[[-5.61951487e-01 5.87241678e-01 1.54909661e-03 ... 3.42299439e-01

3.88963551e-02 -4.99609344e-01]

[-5.61951487e-01 -4.05810961e-01 2.96615118e-01 ... 3.91129021e-01

3.84764709e-01 -4.99609344e-01]

[-5.61951487e-01 5.04487292e-01 3.77087669e-01 ... 1.00963706e+00

2.32582633e-01 -4.99609344e-01]

...

[ 1.77951304e+00 2.45118493e-02 -6.55113628e-02 ... 4.72511657e-01

-1.13285721e-01 -4.99609344e-01]

[ 1.77951304e+00 -8.69235526e-01 7.79450426e-01 ... -4.83372170e-02

5.23112051e-01 -4.99609344e-01]

[ 1.77951304e+00 -3.39607452e-01 6.58741599e-01 ... 6.35276931e-01

8.13641468e-01 -4.99609344e-01]]

*#Splitting Dataset into Training set and Test set*

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

print(X\_train)

[[-0.56195149 0.42173291 0.44414813 ... 0.30974638 0.45393838

-0.49960934]

[-0.56195149 -1.13404956 -1.1653029 ... -1.18769413 -1.04021291

2.00156384]

[-0.56195149 1.62994695 1.154989 ... 1.18867886 1.11800562

-0.49960934]

...

[-0.56195149 -0.83613377 -0.54834667 ... -0.4552504 -0.43148461

-0.49960934]

[-0.56195149 -0.05824254 0.43073604 ... 0.2609168 0.45393838

-0.49960934]

[-0.56195149 -0.24030219 -1.20553917 ... -0.86216358 -1.51059387

-0.49960934]]

print(Y\_train)

[[0.]

[0.]

[0.]

...

[0.]

[0.]

[1.]]

*#Training Model*

classifier = RandomForestClassifier(n\_estimators=100,random\_state=0)

classifier.fit(X\_train,Y\_train)

print(classifier.score(X\_train,Y\_train))

y\_pred = le6.inverse\_transform(np.array(classifier.predict(X\_test),dtype=int))

Y\_test1 = le6.inverse\_transform(np.array(Y\_test,dtype=int))

print(y\_pred)

0.9996875

['No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'Yes' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No'

'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'Yes' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'Yes'

'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'Yes'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'Yes' 'Yes' 'Yes' 'No' 'Yes' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'Yes'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No'

'Yes' 'Yes' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'Yes'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'Yes' 'No' 'No' 'No' 'No' 'Yes' 'Yes' 'No' 'Yes' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No'

'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'Yes' 'Yes' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'Yes' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'Yes' 'Yes'

'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No'

'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'Yes' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No'

'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'Yes' 'Yes' 'No' 'No' 'No' 'No'

'Yes' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'Yes' 'No'

'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'Yes' 'Yes' 'No' 'No' 'Yes' 'No'

'No' 'No' 'No' 'No' 'No' 'Yes' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes'

'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'Yes' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'Yes' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'Yes' 'No' 'Yes' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'Yes' 'No' 'No' 'No' 'No'

'No' 'No' 'Yes' 'Yes']

print(Y\_test1)

['No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No'

'No' 'No' 'Yes' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'Yes'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'Yes'

'No' 'No' 'Yes' 'Yes' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'Yes'

'Yes' 'No' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'Yes' 'No' 'No' 'No' 'No'

'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'Yes' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No'

'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'Yes' 'Yes' 'Yes' 'No'

'No' 'No' 'Yes' 'Yes' 'Yes' 'No' 'Yes' 'No' 'Yes' 'No' 'No' 'No' 'No'

'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'Yes' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No'

'No' 'No' 'Yes' 'No' 'Yes' 'Yes' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'Yes' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'Yes' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'Yes'

'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'Yes' 'Yes' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No'

'No' 'No' 'Yes' 'No' 'No' 'No' 'Yes' 'No' 'Yes' 'No' 'No' 'No' 'Yes'

'Yes' 'No' 'No' 'No' 'No' 'No' 'Yes' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'Yes' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'Yes' 'Yes' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'Yes' 'No'

'Yes' 'Yes' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'Yes' 'No' 'No' 'No' 'Yes' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'Yes' 'Yes' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes'

'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'Yes' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No'

'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'Yes' 'No' 'Yes' 'No' 'No' 'No' 'Yes' 'Yes' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No'

'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No'

'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'Yes' 'No' 'Yes'

'Yes' 'Yes' 'No' 'No' 'Yes' 'No' 'Yes' 'No' 'Yes' 'No' 'Yes' 'No' 'Yes'

'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'Yes' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'Yes' 'Yes' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No'

'Yes' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No'

'No' 'Yes' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'Yes'

'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No'

'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No'

'No' 'No' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes'

'No' 'No' 'No' 'No' 'Yes' 'No' 'Yes' 'Yes' 'No' 'No' 'No' 'Yes' 'No' 'No'

'No' 'No' 'Yes' 'No' 'No' 'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes'

'Yes' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'No' 'Yes' 'No' 'Yes' 'No'

'No' 'Yes' 'No' 'No' 'No' 'Yes' 'Yes']

y\_pred = y\_pred.reshape(-1,1)

Y\_test1 = Y\_test1.reshape(-1,1)

df = np.concatenate((Y\_test1,y\_pred),axis=1)

dataframe = pd.DataFrame(df,columns=['Rain on Tommorrow','Prediction of Rain'])

print(dataframe)

Rain on Tommorrow Prediction of Rain

0 No No

1 No No

2 No No

3 No No

4 No No

.. ... ...

795 No No

796 No No

797 No No

798 Yes Yes

799 Yes Yes

[800 rows x 2 columns]

Bagging Classifier Accuracy: 0.8775

**from** **sklearn.ensemble** **import** GradientBoostingClassifier

dt = GradientBoostingClassifier(n\_estimators=170,max\_depth=1)

dt.fit(X\_train,Y\_train)

print(dt.score(X\_train,Y\_train))

y\_pred = le6.inverse\_transform(np.array(dt.predict(X\_test),dtype=int))

Y\_test3 = le6.inverse\_transform(np.array(Y\_test,dtype=int))

y\_pred = y\_pred.reshape(-1,1)

Y\_test3 = Y\_test3.reshape(-1,1)

df = np.concatenate((Y\_test3,y\_pred),axis=1)

dataframe = pd.DataFrame(df,columns=['Rain on Tommorrow','Prediction of Rain'])

print(dataframe)

dt\_accuracy = accuracy\_score(Y\_test1,y\_pred)

print("**\n**Gradient Boosting Accuracy: "+str(dt\_accuracy))

*#print(y\_pred)*

*#print(Y\_test)*

0.878125

Rain on Tommorrow Prediction of Rain

0 No No

1 No No

2 No No

3 No No

4 No No

.. ... ...

795 No No

796 No Yes

797 No No

798 Yes Yes

799 Yes Yes

[800 rows x 2 columns]

Gradient Boosting Accuracy: 0.88

**import** **xgboost** **as** **xgb**

xg = xgb.XGBClassifier(n\_estimators=140, max\_depth=12)

xg.fit(X\_train,Y\_train)

print(xg.score(X\_train,Y\_train))

y\_pred = le6.inverse\_transform(np.array(xg.predict(X\_test),dtype=int))

Y\_test4 = le6.inverse\_transform(np.array(Y\_test,dtype=int))

*#print(y\_pred)*

*#print(Y\_test)*

y\_pred = y\_pred.reshape(-1,1)

Y\_test4 = Y\_test4.reshape(-1,1)

df = np.concatenate((Y\_test4,y\_pred),axis=1)

dataframe = pd.DataFrame(df,columns=['Rain on Tommorrow','Prediction of Rain'])

print(dataframe)

dt\_accuracy = accuracy\_score(Y\_test1,y\_pred)

print("**\n**XGBoost Accuracy: "+str(dt\_accuracy))

1.0

Rain on Tommorrow Prediction of Rain

0 No No

1 No No

2 No No

3 No No

4 No No

.. ... ...

795 No No

796 No No

797 No No

798 Yes Yes

799 Yes No

[800 rows x 2 columns]

XGBoost Accuracy: 0.875

height = dt\_accuracy

bars = ('RFT',' Bagging', 'GradientBoosting' ,'XGBoost')

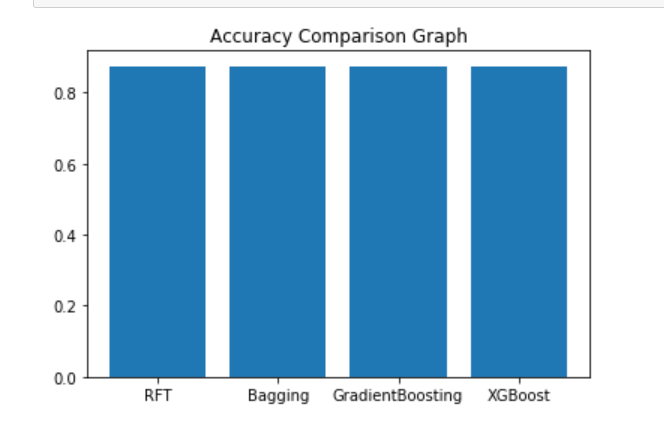
y\_pos = np.arange(len(bars))

plt.bar(y\_pos, height)

plt.xticks(y\_pos, bars)

plt.title("Accuracy Comparison Graph")

plt.show()



**ScreenShots :**

**Packages:**

Pip is the package management system and is used to install software packages. The packages that are being installed and used are :

packages details

pip install pandas==0.25.3

pip install numpy==1.19.2

pip install scikit-learn==0.22.2.post1

pip install xgboost==1.0.2

pip install jupyter==1.0.0

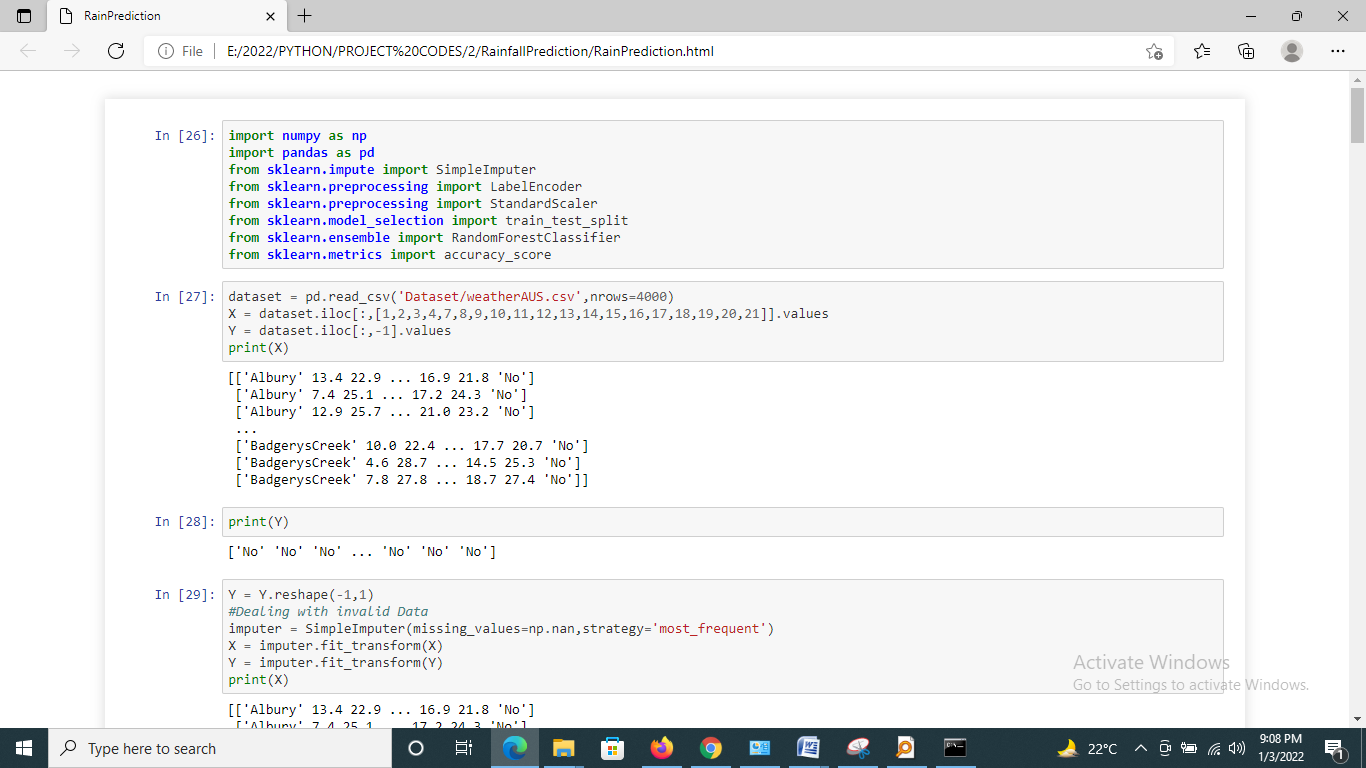
pip install   jupyter-client==6.1.3

pip install   jupyter-console==6.4.0

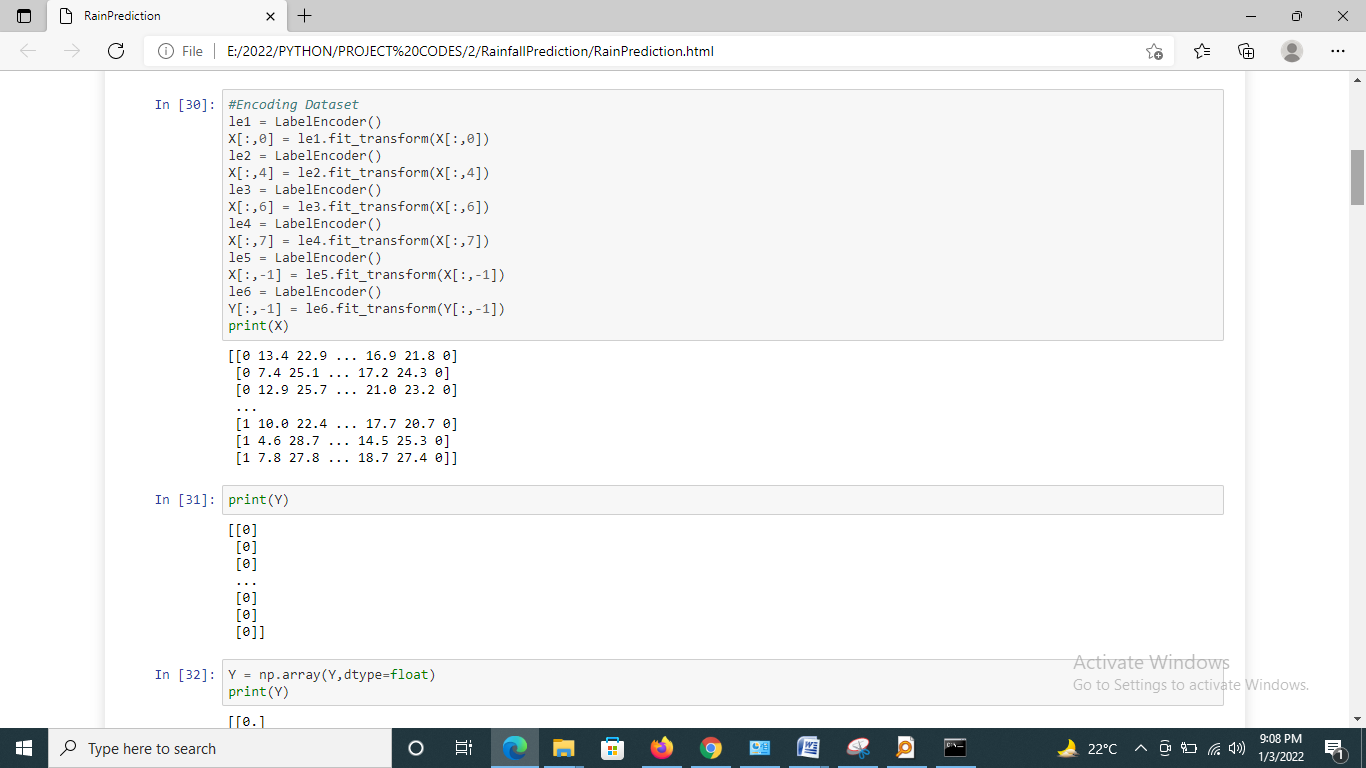
pip install   jupyter-core==4.6.3

pip install   jupyterlab-widgets==1.0.0

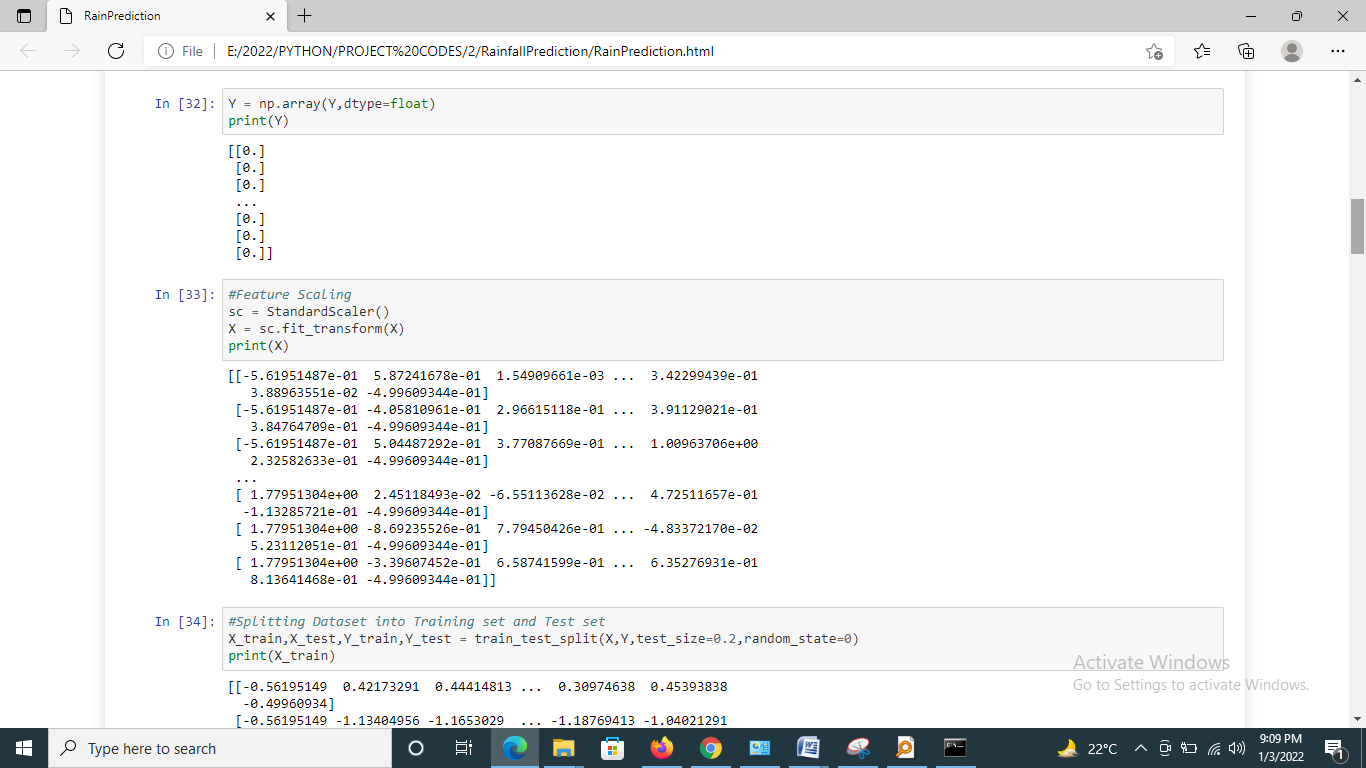
While random forest, gradient boosting and bagging classifier consists in scikit, xgboost is a separate library which must be installed using pip.

****

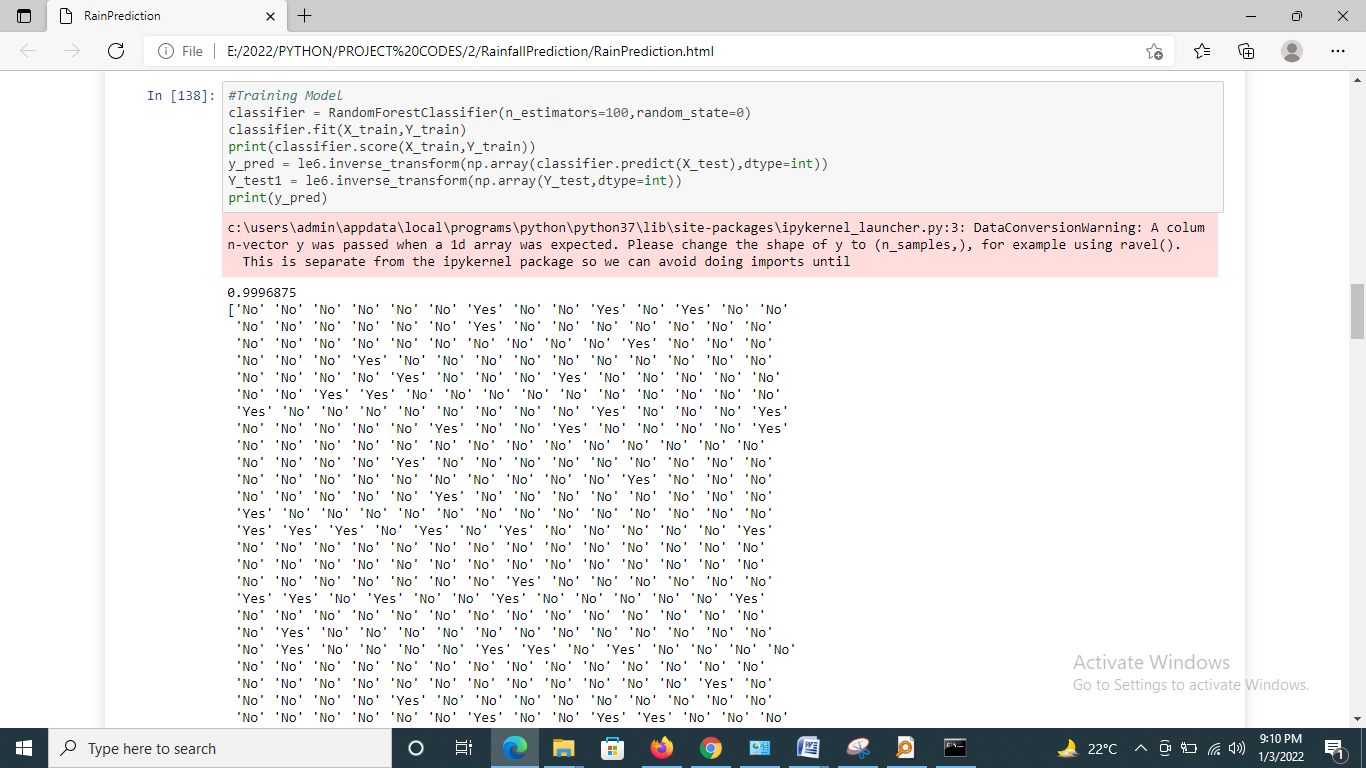
**Analysis**

****

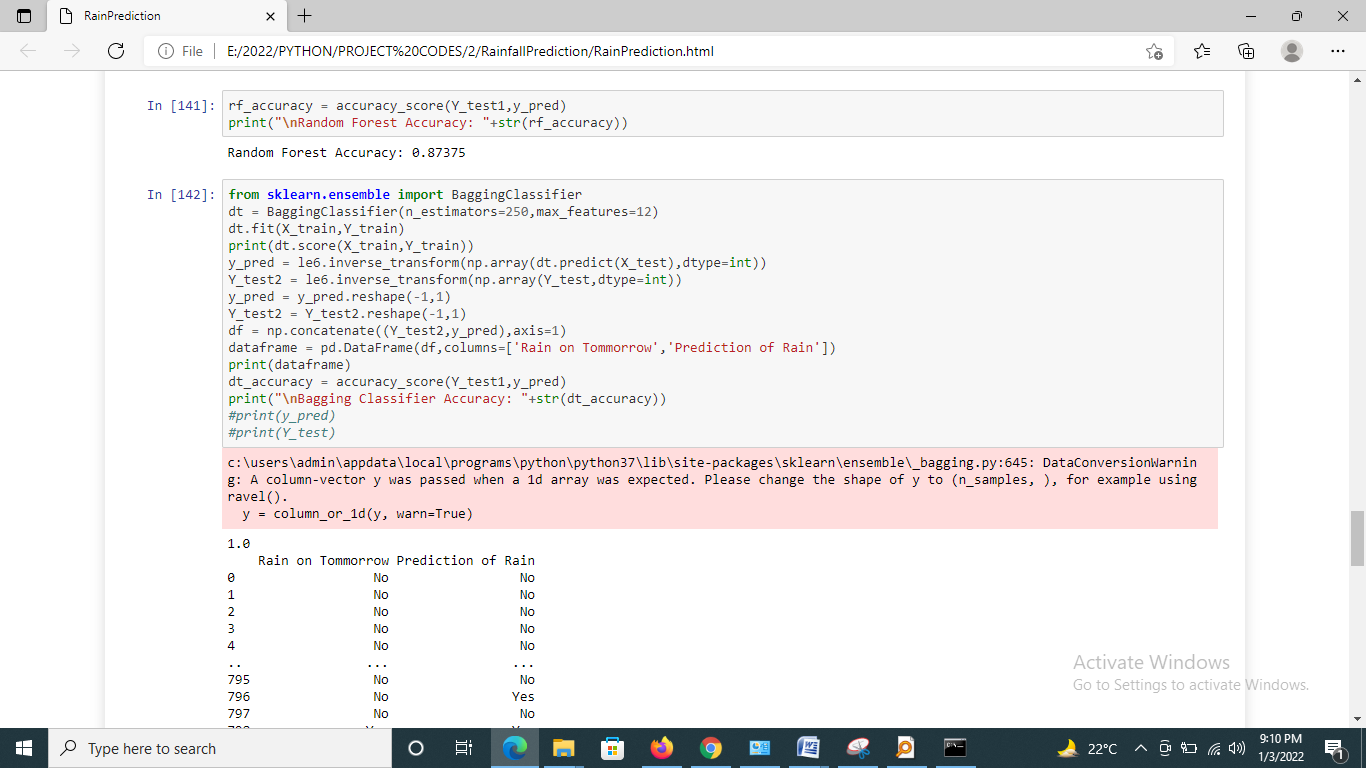
**Training**

****

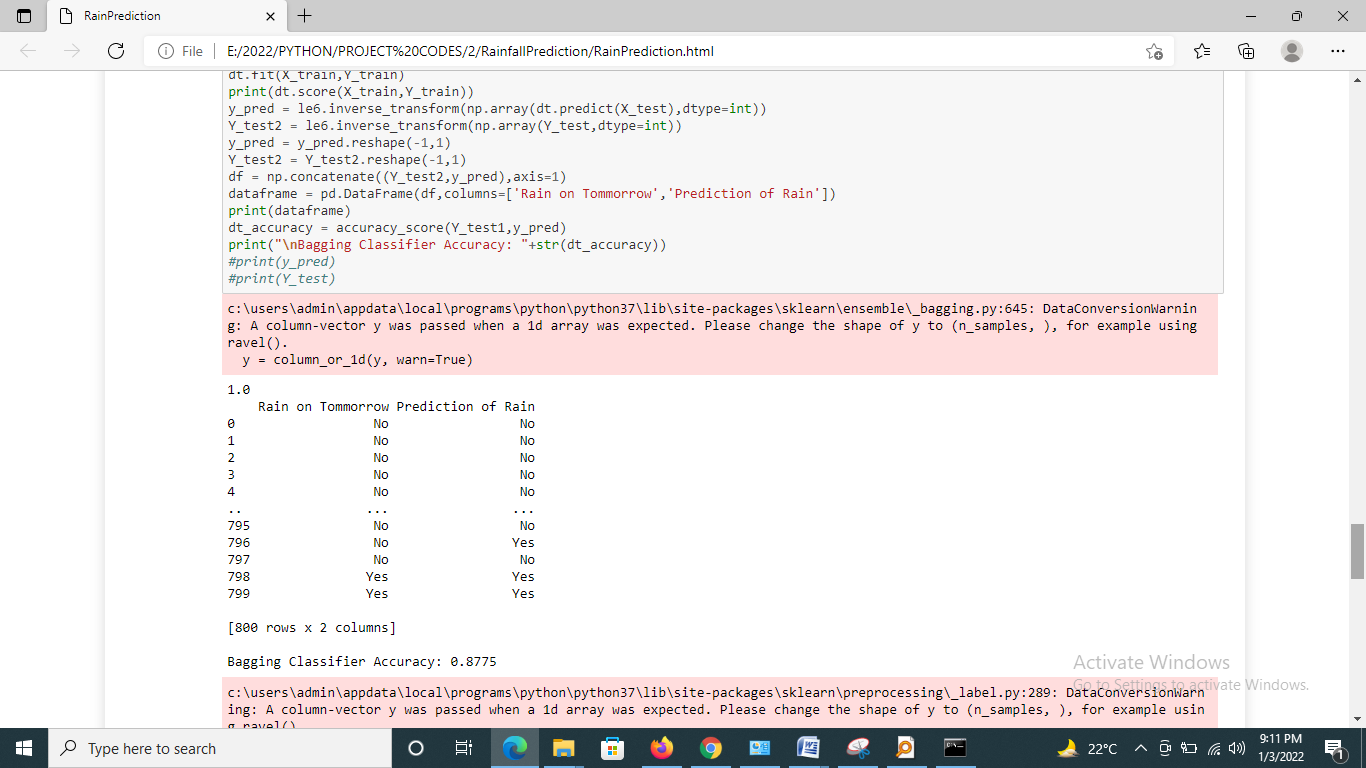
**Algorithms:**

****

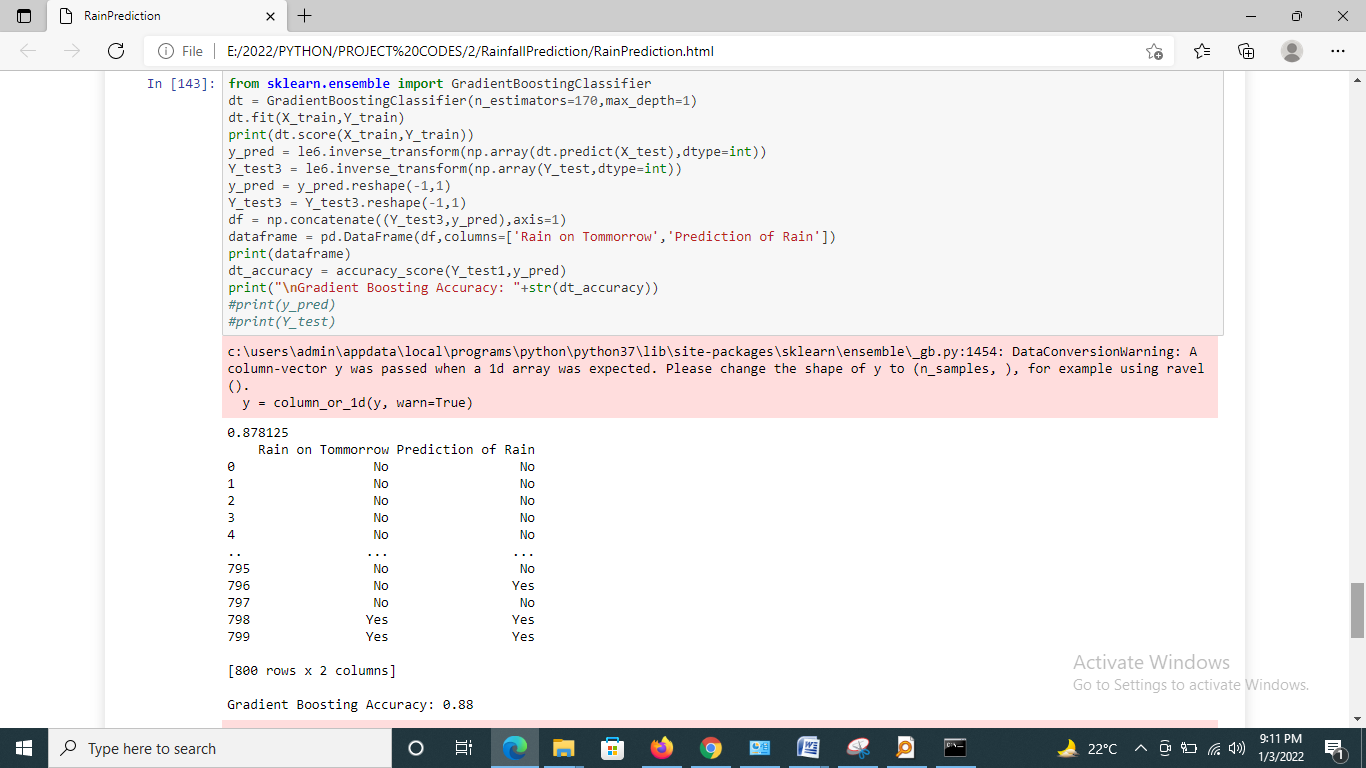
**Random Forest :**

****

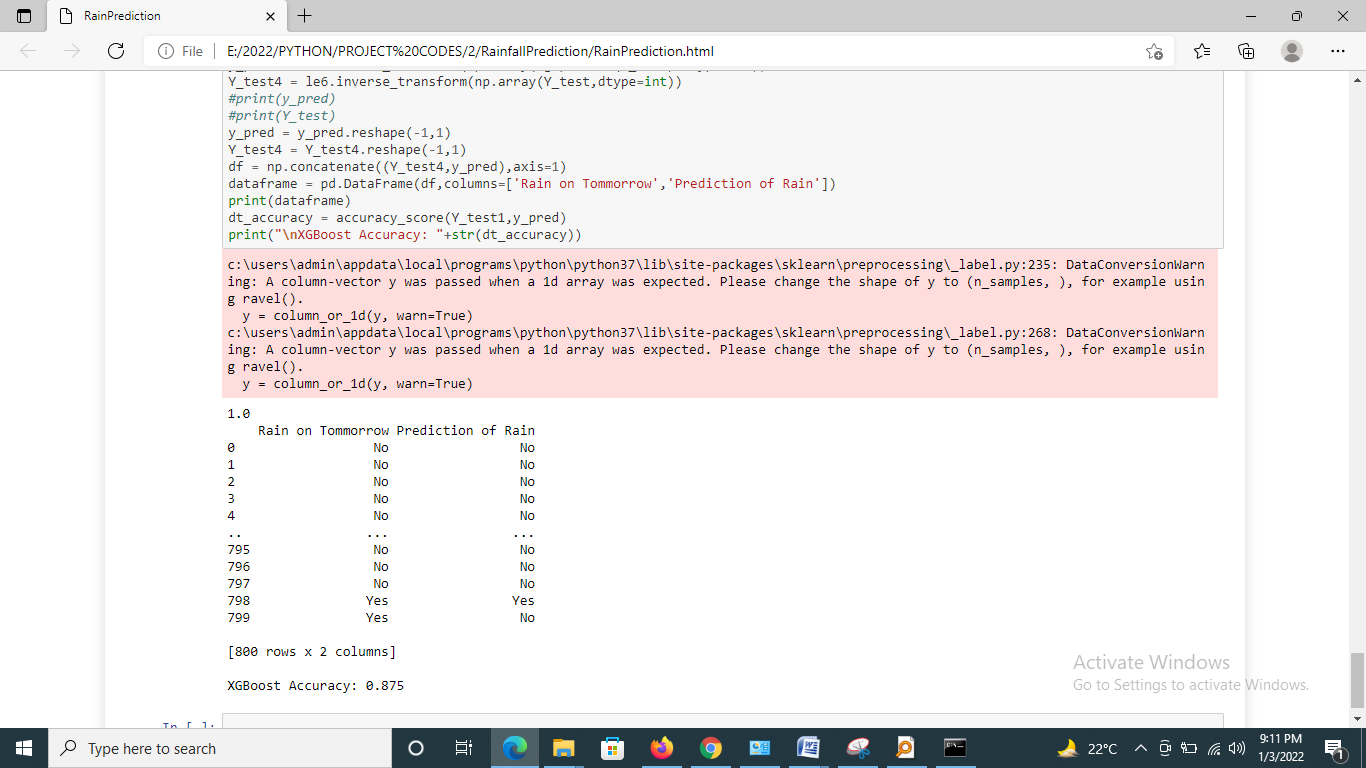
**Bagging Classifier:**

****

**Gradient Boosting:**

****

**Xgboost :**

****

**Conclusion and Future Work**

In this paper, we explored and applied several preprocessing steps and learned there impact on the overall performance of our classifiers. We also carried a comparative study of all the classifiers with different input data and observed how the input data can affect the model predictions.

We can conclude that Australian weather is uncertain and there is no such correlation among rainfall and the respective region and time. We figured certain patterns and relationships among data which helped in determining important features. Refer to the appendix section. As we have a huge amount of data, we can apply Deep Learning models such as Multilayer Perceptron, Convolutional Neural Network, and others.

The study aimed at building a predicting system using random forest and svm that could predict monthly rainfall accurately and efficiently with minimum error and the rainfall predictions after training, testing are obtained that are quite accurate and through comparison outlined that the actual and predicted data for these areas illustrated finest results using the certainly different parameters of the rainfall that are different for different areas with minimum error observed. This project concentrated on estimation of rainfall and it is estimated that Bagging and Boosting techniques have a valuable and adaptable strategy, helping to manage the impediments relating to distributional properties of fundamental factors, geometry of the information and the normal issue of model over fitting.

It would be great to perform a comparative study between the Machine learning classifiers and Deep learning models.

**References :**

1. World Health Organization: Climate Change and Human Health: Risks and Responses. World Health Organization, January 2003

2. Alcntara-Ayala, I.: Geomorphology, natural hazards, vulnerability and prevention of natural disasters in developing countries. Geomorphology 47(24), 107124 (2002)

3. Nicholls, N.: Atmospheric and climatic hazards: Improved monitoring and prediction for disaster mitigation. Natural Hazards 23(23), 137155 (2001)

4. [Online] InDataLabs, Exploratory Data Analysis: the Best way to Start a Data Science Project. Available: https://medium.com/@InDataLabs/ why-start-a-data-science-project-with-exploratory-data-analysis-f90c0efcbe49

5. [Online] Pandas Documentation. Available: https://pandas.pydata.org/ pandas-docs/stable/reference/api/pandas.get\\_dummies.html

6. [Online] Sckit-Learn Documentation Available: https://scikit-learn.org/ stable/modules/generated/sklearn.feature\\_extraction.FeatureHasher. html

7. [Online] Sckit-Learn Documentation Available: https://scikit-learn.org/ stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

8. [Online] Sckit Learn Documentation Available: https://scikit-learn.org/ stable/modules/generated/sklearn.feature\_selection.SelectKBest.html

9. [Online] Raheel Shaikh, Feature Selection Techniques in Machine Learning with Python Available: https://towardsdatascience.com/ feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e

10. [Online] Imbalanced-learn Documentation Available: https://imbalanced-learn. readthedocs.io/en/stable/introduction.html

11. V. Veeralakshmi and D. Ramyachitra, Ripple Down Rule learner (RIDOR) Classifier for IRIS Dataset. Issues, vol 1, p. 79-85.

12. [Online] Aditya Mishra, Metrics to Evaluate your Machine Learning Algorithm Available: <https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234>

13. Mosavi, A., Ozturk, P., & Chau, K. W. (2018). Flood prediction using machine learning models: Literature review.10(11). https://doi.org/10.3390/w10111536 2. Janani, B; Sebastian, P. (2014). Analysis on the weather forecasting and techniques. International Journal of Advanced Research in Computer Engineering & Technology, 3(1), 59–61. http://ijarcet.org/wp-content/uploads/IJARCET-VOL-3-ISSUE1-59- 61.pdf

14.https://www.researchgate.net/publication/336914968\_Predicting\_Rainfall\_using\_Machine\_Learning\_Techniques/link/6075ad4 992851cb4a9dbee71/download

15.https://www.analyticsvidhya.com/blog/2020/04/feature-scalingmachine-learning-normalization-standardization

16. http://www.ijstr.org/final-print/jan2020/Prediction-Of-RainfallUsing-Machine-Learning-Techniques.pdf 9. KavithaRani, B., & Govardhan, A. (2014). Effective Features and Hybrid Classifier for Rainfall Prediction. International Journal of Computational Intelligence Systems.