



FUTURE INSTITUTE OF ENGINEERING AND MANAGEMENT

**In association with**



(ISO9001:2015)

## **WEATHER FORECASTING**

**A PROJECT PRESENTATION**

### **PRESENTED BY :-**

1. RUPAK CHAKRABORTY
2. ANKIT SINHA
3. SHRAMANA BOSE
4. AMEYA KUMAR NATH

### **UNDER THE GUIDANCE OF :**

**MAHENDRA DATTA**

# INTRODUCTION

- *Weather forecasting plays a crucial role in our daily lives, impacting activities ranging from agriculture and transportation to disaster preparedness. Traditionally, weather predictions have relied on numerical weather models and meteorological observations. However, in recent years, machine learning (ML) has emerged as a powerful tool to enhance the accuracy and efficiency of weather forecasting. But losses could be minimize by making the adjustment with coming weather through timely and accurate weather forecasting.*
- *The current weather prediction models heavily depend on complex physical models and need to be run on large computer systems involving hundreds of HPC nodes. The computational power of these large systems is required to solve the models that describe the atmosphere. Despite using these costly and complex devices, there are often inaccurate forecasts because of incorrect initial measurements of the conditions or an incomplete understanding of atmospheric processes. Moreover, it generally takes a long time to solve complex models like these.*
- *As weather systems can travel a long way over time in all directions, the weather of one place depends on that of others considerably [10]. In this work, we propose a method to utilize surrounding city's historical weather data along with a particular city's data to predict its weather condition. We combine these data and use it to train simple machine learning models, which in turn, can predict correct weather conditions for the next few days. These simple models can be run on low cost and less resource-intensive computing systems, yet can provide quick and accurate enough forecasts to be used in our day-to-day life.*

# LITERATURE REVIEW

---

*There are many research papers that have been published related to predicting the weather.*

- A paper was published on 'The Weather Forecast Using Data Mining Research Based on Cloud Computing' This paper proposes a modern method to develop a service oriented architecture for the weather information systems which forecast weather using these data mining techniques. This can be carried out by using Artificial Neural Network and Decision tree Algorithms and meteorological data collected in Specific time. Algorithm has presented the best results to generate classification rules for the mean weather variables. The results showed that these data mining techniques can be enough for weather forecasting.*
- Another paper was published on 'Analysis on The Weather Forecasting and Techniques' where they decided that artificial neural network and concept of fuzzy logic provides a best solution and prediction comparatively . They decided to take temperature, humidity, pressure, wind and various other attributes into consideration. Another research paper titled 'Issues with weather prediction' discussed the major problems with weather prediction. Even the simplest weather prediction is not perfect. The one-day forecast typically falls within two degrees of the actual temperature. Although this accuracy isn't bad, as predictions are made for further in time. For example, in a place like New England where temperatures have a great variance the temperature prediction are more inaccurate than a place like the tropics. Another research paper titled 'Current weather prediction' used numerical methods to stimulate what is most likely going to happen based on known state of the atmosphere. For example, if a forecaster is looking at three different numerical models, and two model predict that a storm is going to hit a certain place, the forecaster would most likely predict that the storm is going to hit the area. These numerical models work well and are being tweaked all the time, but they still have errors because some of the equations used by the models aren't precise.*

# MACHINE LEARNING IN WEATHER FORECASTING

Machine learning (ML) has shown promising results in various fields, including weather forecasting. Traditional weather forecasting relies on numerical weather prediction models that simulate the Earth's atmosphere based on physical principles. However, these models have limitations due to the complex and chaotic nature of the atmosphere. Machine learning techniques can complement these traditional methods and improve the accuracy of weather predictions. Here are some ways in which machine learning is used in weather forecasting:

## ❑ Data Collection and Preprocessing:

Gathering diverse meteorological data, including satellite imagery, weather station reports, radar data, and atmospheric measurements.

Cleaning and preprocessing the data to ensure consistency, removing noise, and addressing missing or erroneous information.

## ❑ Feature Extraction:

Identifying relevant features from the data that contribute to weather patterns, such as temperature, humidity, wind speed, and atmospheric pressure.

Extracting spatial and temporal features to capture the dynamic nature of weather conditions.

## ❑ Model Training:

Utilizing machine learning algorithms, such as neural networks, decision trees, or ensemble methods, to train models on historical weather data.

Models learn patterns and relationships within the data, allowing them to make predictions based on input features.

## ❑ Ensemble Forecasting:

Creating ensemble models that consider multiple scenarios or variations in initial conditions to account for uncertainties in weather predictions.

Combining the outputs of different models to generate more reliable and robust forecasts.

## ❑ Post-processing and Calibration:

Applying machine learning techniques for post-processing of numerical weather prediction outputs.

Calibrating model forecasts to correct biases and improve the overall accuracy of predictions.

## ❑ Continuous Improvement:

Iteratively refining machine learning models by incorporating new data and updating algorithms to adapt to changing climate patterns.



# METHODOLOGY

In our project, we are trying to compare the different machine learning technique use for predicting the events. Machine learning models are designed to predict the events on the basis of the Precip -type, Temperature, Apparent Temperature , pressure, Humidity, Visibility, Wind Sped ,summary. In this Project, we have taken the weather data from kaggle ,to train our machine learning model.

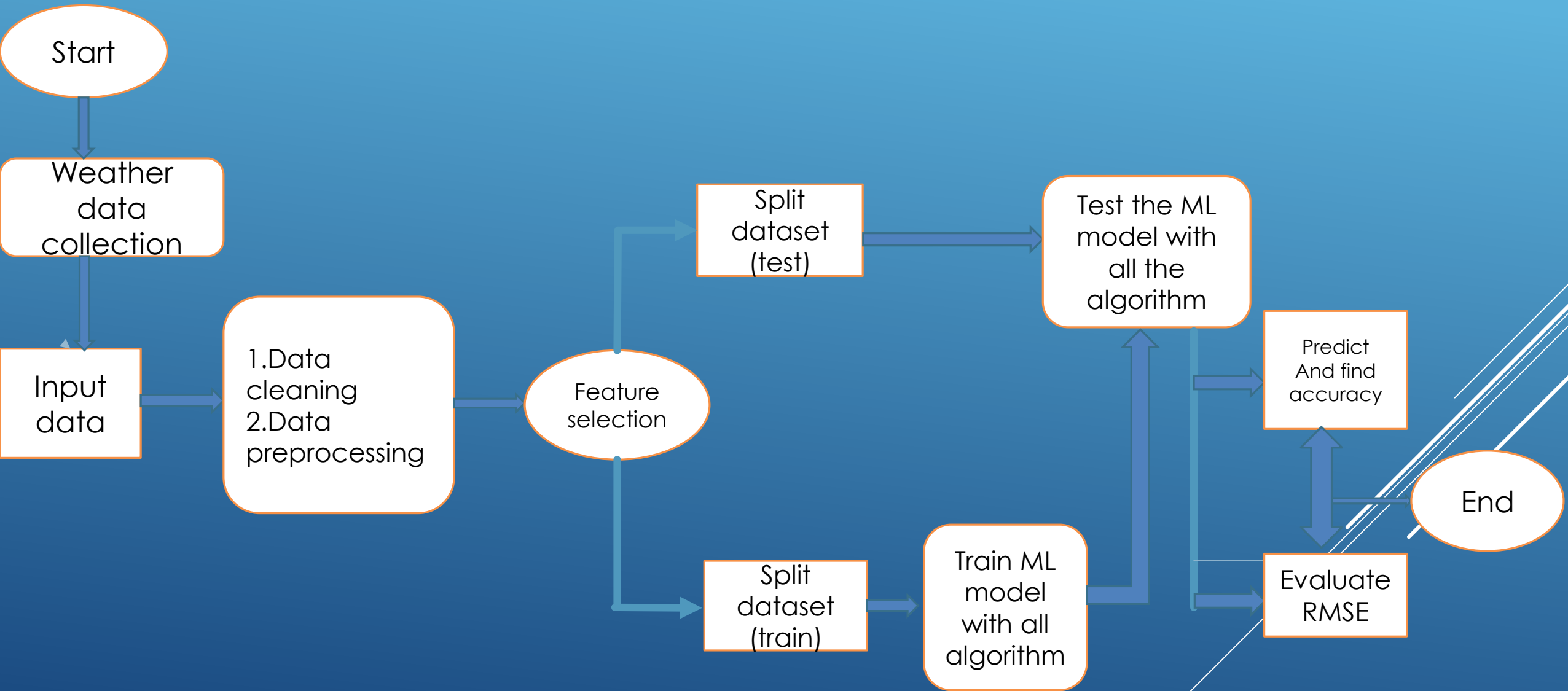
Here, we are using random forest, decision tree, support vector regression, KNN, naïve Bayes and linear regression to evaluate and train our model.

- ❖ Initially, we preprocessed the dataset by eliminating any incomplete record. We apply feature selection and eliminate features that have no direct impact on the performance.
- ❖ Then we have to encode the categorical features into numerical feature because our machine learning algorithms will not be able to understand the categorical value. Here when we try to encode the categorical feature into numerical using label encoder our model not showed a good accuracy score that's why here we used ordinal encoding method to encode the categorical features
- ❖ It is necessary scale the numerical value for better performance of our ML model .here we use standard scaler method.

- ❖ We also checked that whether we have outliers . And also we took a look at the multi collinearity and drop some highly co-related columns because if we keep some highly correlated features means it will affect our model.
- ❖ Once our dataset is split using train test split on 80:20 ratio, the predictive models like linear regression, random forest, decision tree, support vector machine and k-nearest neighbor used to forecast upcoming match results.



# *Machine Learning process overview*



# *DATASET*

*Proposed system is implemented using the Weather history dataset from Kaggle. The WeatherHistory.csv file contains water quality metrics for 96453 dataset, 9 features and one class variable.*

*Feature list :*

- 1. Summary*
- 2. Precip Type*
- 3. Temperature (C)*
- 4. Apparent Temperature (C)*
- 5. Humidity*
- 6. Wind Speed (km/h)*
- 7. Wind Bearing (degrees)*
- 8. Visibility (km)*
- 9. Pressure (mill bars)*

## DIFFIERENT ML ALGORITHM FOR PREDICTING THE VALUE

### □ K-NEAREST NEIGHBORS (KNN) :-

KNN is a supervised learning method that may be used for both regression and classification, however it is usually utilized for classification. KNN aims to predict the right class of testing data given a set with various classes by calculating the distance between both the testing data and all the training points. It then chooses the k points that are the most similar to the test.

After the points have been chosen, the algorithm calculates the likelihood (in the classification phase) that the test point belongs to one of the k training point classes, and the class with the greatest probability is chosen.

```
from sklearn.neighbors import KNeighborsClassifier
model=KNeighborsClassifier(weights="distance")
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
from sklearn.metrics import accuracy_score
print("KNN CLASSIFIER:-",accuracy_score(y_test,y_pred))
mse=mean_squared_error(y_pred,y_test)
print("mse:-","{:.4}".format(mse))
import numpy as np
r_mse=np.sqrt(mse)
print("r_mse:-","{:.4}".format(r_mse))
```

```
KNN CLASSIFIER:- 0.98186123208321
mse:- 0.02461
r_mse:- 0.1569
```

- THE ACCURACY OFF KNN CLASSIFIER: 98.1%
- MEAN SQUARED ERROR: 0.02461
- ROOT MSE: 0.1569



## ❑ SUPPORT VECTOR MACHINE:-

Support Vector Machine (SVM) is a powerful machine learning algorithm used for linear or nonlinear classification, regression, and even outlier detection tasks. SVMs can be used for a variety of tasks, such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection. SVMs are adaptable and efficient in a variety of applications because they can manage high-dimensional data and nonlinear relationships.

SVM algorithms are very effective as we try to find the maximum separating hyper plane between the different classes available in the target feature.

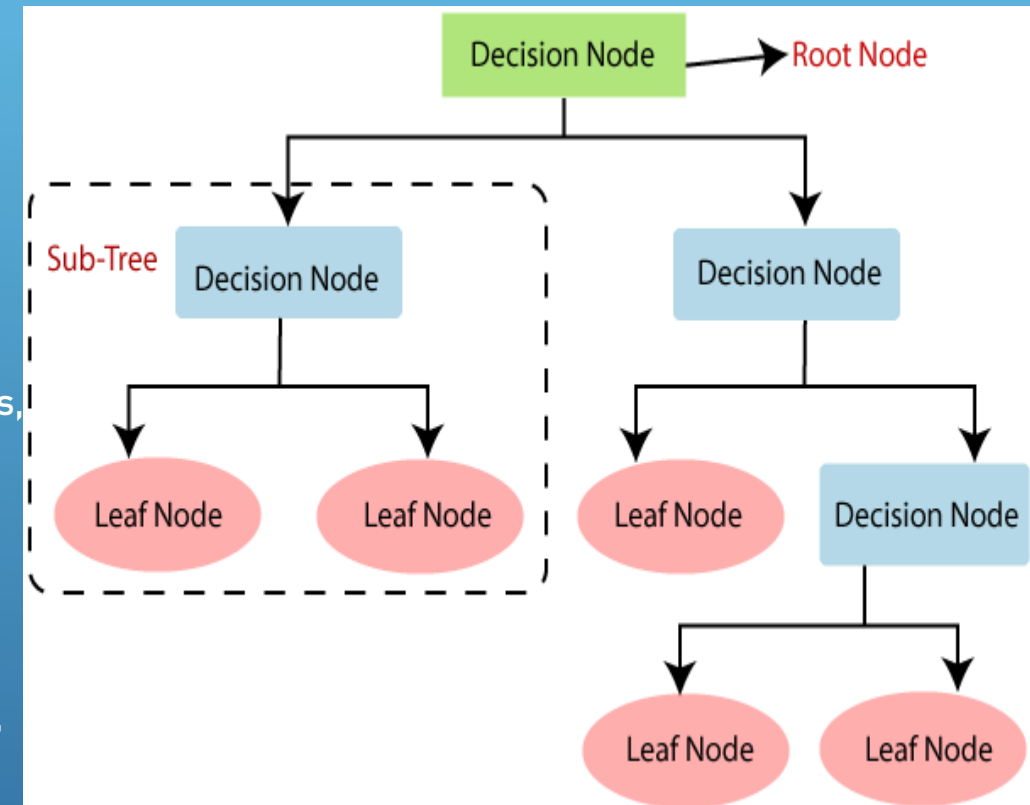
```
from sklearn.svm import SVC
svc_model=SVC(kernel="linear")
svc_model.fit(x_train,y_train)
y_pred=svc_model.predict(x_test)
from sklearn.metrics import accuracy_score
print("SVC linear=",accuracy_score(y_test,y_pred))
mse=mean_squared_error(y_pred,y_test)
print("mse:-","{:.4}".format(mse))
import numpy as np
r_mse=np.sqrt(mse)
print("r_mse:-","{:.4}".format(r_mse))
```

```
SVC linear= 0.9999154369794089
mse:- 0.0002114
r_mse:- 0.01454
```

- THE ACCURACY OFF KNN CLASSIFIER: 99%
- MEAN SQUARED ERROR : 0.0002114
- ROOT MSE : 0.01454

## ❑ DECISION TREE CLASSIFIER : -

- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
- In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- *It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.*
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.



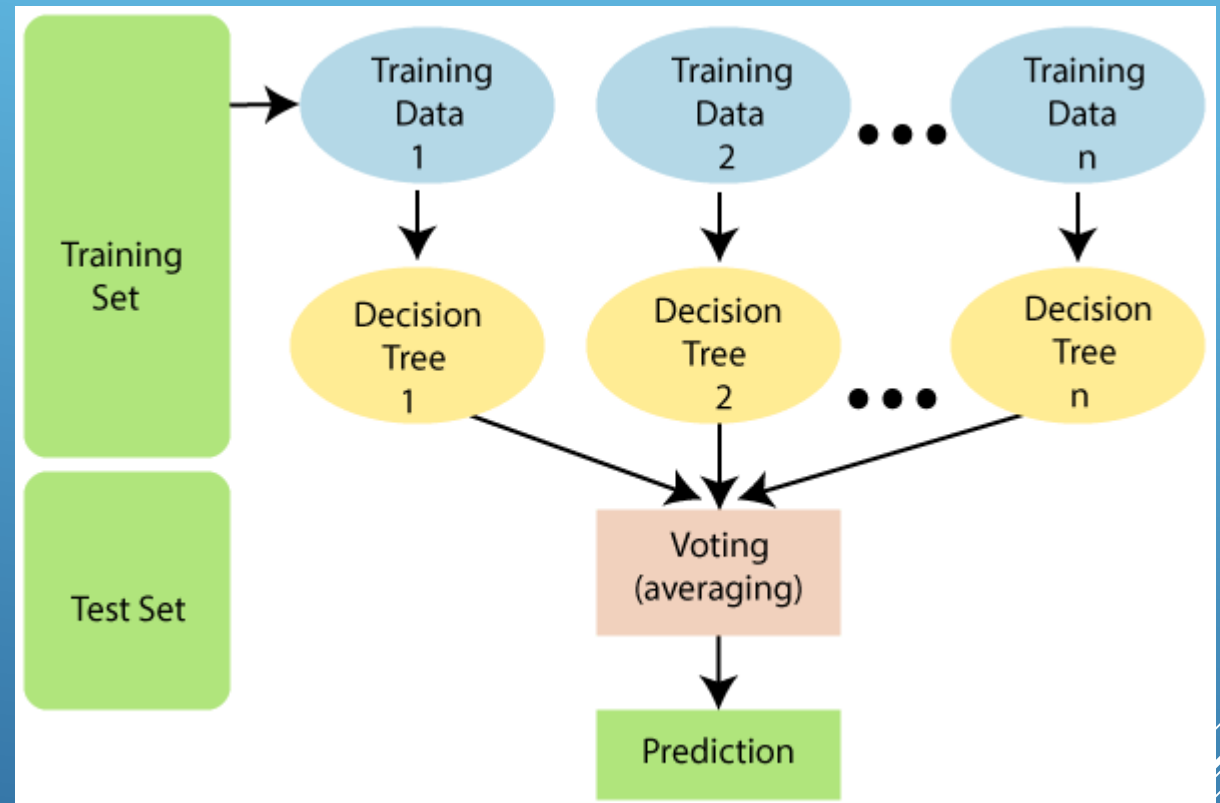
```
from sklearn.tree import DecisionTreeClassifier
model_tree=DecisionTreeClassifier(criterion="gini",random_state=42)
model_tree.fit(x_train,y_train)
y_pred_tree=model_tree.predict(x_test)
from sklearn.metrics import accuracy_score
print("Decision_tree_classifier:-", accuracy_score(y_test,y_pred_tree))
mse=mean_squared_error(y_pred,y_test)
print("mse:-","{:.4}".format(mse))
import numpy as np
r_mse=np.sqrt(mse)
print("r_mse:-","{:.4}".format(r_mse))
```

```
Decision_tree_classifier:- 0.9998731554691134
mse:- 0.02461
r_mse:- 0.1569
```

- THE ACCURACY OFF DECISION TREE CLASSIFIER: 99%
- MEAN SQUARED ERROR : 0.02461
- ROOT MSE : 0.1569

## ❑ RANDOM FOREST CLASSIFIER :-

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model*. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of over fitting.



```
from sklearn.ensemble import RandomForestClassifier
model_rf=RandomForestClassifier(n_estimators=20,criterion="entropy",random_state=0)
model_rf.fit(x_train,y_train)
y_pred_rf=model_rf.predict(x_test)
from sklearn.metrics import accuracy_score
print("random forest classifier:-",accuracy_score(y_test,y_pred_rf))
mse=mean_squared_error(y_pred_rf,y_test)
print("mse:-","{:.4}".format(mse))
import numpy as np
r_mse=np.sqrt(mse)
print("r_mse:-","{:.4}".format(r_mse))
```

```
random forest classifier:- 0.9993657773455668
mse:- 0.005327
r_mse:- 0.07299
```

- THE ACCURACY OF RANDOM FOREST CLASSIFIER: 99%
- MEAN SQUARED ERROR: 0.005327
- ROOT MSE : 0.07299

## ❑ Naive Bayes Classifier :-

Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems. It is mainly used in *text classification* that includes a high-dimensional training dataset.

Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.

It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.

Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles

Why is it called Naïve Bayes?

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, Which can be described as:

Naïve: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features.

Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple.

Hence each feature individually contributes to identify that it is an apple without depending on each other.

Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem.

Bayes' theorem is also known as Bayes' Rule or Bayes' law, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.

The formula for Bayes' theorem is given as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

```
from sklearn.naive_bayes import GaussianNB
model_nb=GaussianNB()
model_nb.fit(x_train,y_train)
y_pred_2=model_nb.predict(x_test)
from sklearn.metrics import accuracy_score
print("GAUSSIAN_CLASS:-",accuracy_score(y_test,y_pred_2))
mse=mean_squared_error(y_pred_2,y_test)
print("mse:-","{:.4}".format(mse))
import numpy as np
r_mse=np.sqrt(mse)
print("r_mse:-","{:.4}".format(r_mse))
```

```
GAUSSIAN_CLASS:- 0.9999577184897045
mse:- 4.228e-05
r_mse:- 0.006502
```

- THE ACCURACY OF NAIVE BAYES CLASSIFIER: 99%
- MEAN SQUARED ERROR : 4.228e
- ROOT MSE : 0.006502

## ❑ LINEAR REGRESSION ALGORITHM :-

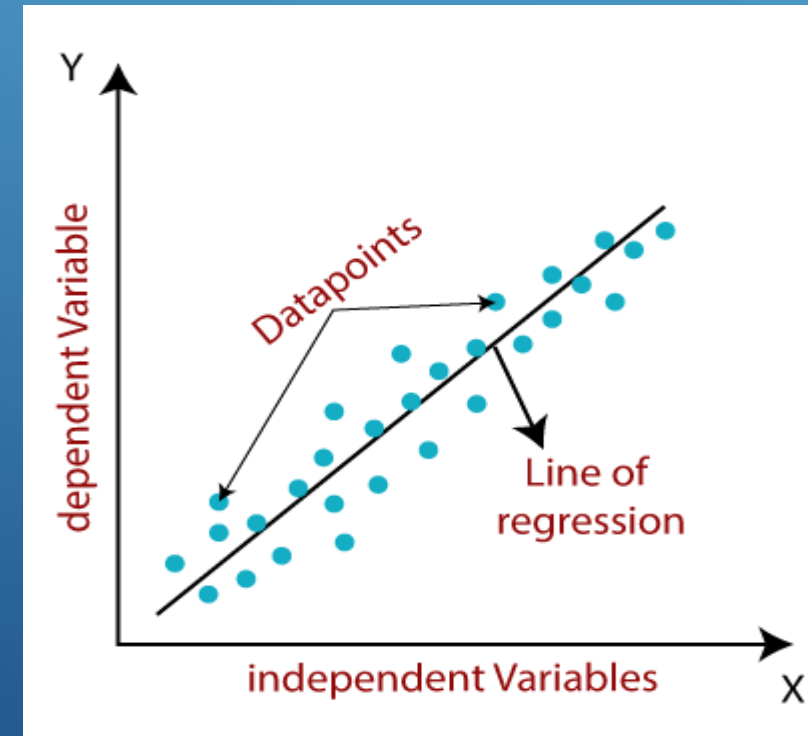
Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc.

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (x) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable. The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:

```
from sklearn.linear_model import LinearRegression
model_lr=LinearRegression()
model_lr.fit(x_train,y_train)
y_pred_lr=model_lr.predict(x_test)
from sklearn.metrics import mean_squared_error
mse=mean_squared_error(y_pred_lr,y_test)
print("mse:-","{:.4}".format(mse))
import numpy as np
r_mse=np.sqrt(mse)
print("r_mse:-","{:.4}".format(r_mse))

mse:- 1.051e-27
r_mse:- 3.242e-14
```

- MEAN SQUARED ERROR: 1.051e
- ROOT MSE : 3.242e





# Conclusion

*The weather prediction done using linear regression algorithm, Random forest algorithm and Naïve Bayes algorithm are very essential for improving the future performance for the people. For predicting the weather, the linear regression algorithm and Naïve Bayes algorithm, Random forest, Decision tree, K-nearest neighbors was applied to the datasets of the weather. We made a model to predict the weather using some selected input variables collected from Kaggle. The problem with current weather scenario is that we are not able to prepare our self and not able to do some important works. So, for knowing the weather scenario at high accuracy considering every factor that affects in the weather scenario, this model is Therefore in this we provided how the machine learning techniques can be trained and used for the weather forecasting. In this Machine learning models are much accurate than human prediction and physical models prepared by human. Accuracy obtained here was measured on the basis of coefficient correlation. We also utilize the historical data to predict the weather conditions which is much faster than the traditional models. The new pattern is combining deterministic and machine learning or statistical components, can provide fast and accurate calculations of these processes as well and help in predicting value of independent variable accurately. In future work, going to do research and make a model on how the neighboring weather can affect the weather of our area*

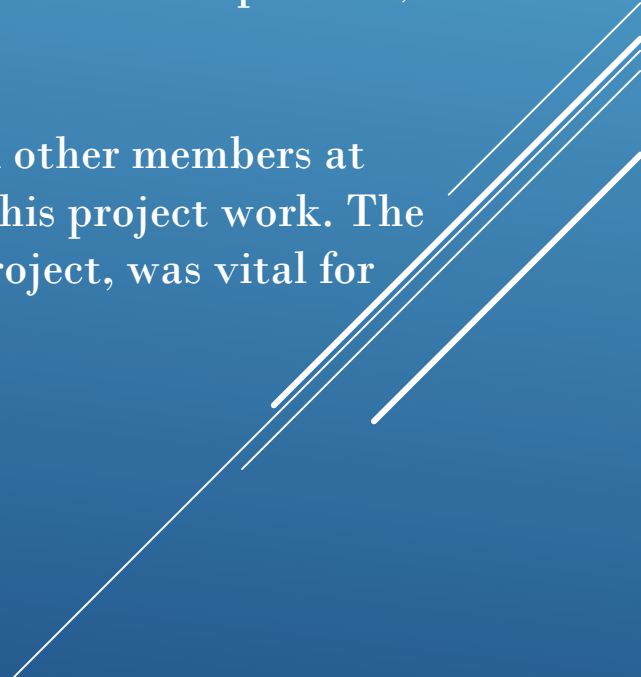


# Acknowledgement

Success of any project depends largely on the encouragement and guidelines of many others. I take this sincere opportunity to express my gratitude to the people who have been instrumental in the successful completion of this project work.

I would like to show our greatest appreciation to Mr. MAHENDRA DUTTA, Project Engineer at Ardent, Kolkata. I always feel motivated and encouraged every time by his valuable advice and constant inspiration; without his encouragement and guidance this project would not have materialized.

Words are inadequate in offering our thanks to the other trainees, project assistants and other members at Ardent Computech Pvt. Ltd. for their encouragement and cooperation in carrying out this project work. The guidance and support received from all the members and who are contributing to this project, was vital for the success of this project.

Several white lines of varying lengths and angles are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

*THANK YOU....*

