GIET UNIVERSITY,GUNUPUR

Dist. - Rayagada, Odisha-765022, Contact: - +91 7735745535, 06857-250170,172, Visit us:- www.giet.edu

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

A PROJECT REPORT ON
"RAINFALL PREDICTION SYSTEM"

Submitted in the partial fulfillment of the requirements in the 5th semester of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

BY

PIYUSH KUMAR (20CSE038)
ANURAG THAKUR (20CSE036)
PRIYA KHETAN (20CSE052)

Under the supervision of
Dr.AVS Pawan Kumar

GIETUNIVERSITY,GUNUPUR

Dist. - Rayagada, Odisha-765022, Contact:- +91 7735745535, 06857-250170,172, Visit us:- www.giet.edu

---

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project work entitled "Rainfall Prediction" is done by Piyush Kumar(20CSE038), Anurag Thakur(20CSE036) and Priya Khetan(20CSE052) bearing Regd. No.- 20UG010171, 20UG010169 and 20UG010184 in partial fulfillment of the requirements for the 5th Semester Sessional Examination of Bachelor of Technology in Computer Science and Engineering during the academic year 2022-23. This work is submitted to the department as a part of evaluation of 5th Semester Minor Project-I.

Class Teacher: Dr. Bidush Kumar Sahoo
Project Supervisor: Dr.AVS Pawan Kumar
Project Coordinator: Mr. Bhavani Shankar Panda

Project Coordinator | Class Teacher | Project Supervisor

ACKNOWLEDGEMENT

First and foremost, I would like to thank our project supervisor Dr.AVS Pawan Kumar for supporting and guiding us in project.

I would like to thank our class teacher Dr. Bidush Kumar Sahoo for his constant encouragement and support extended towards completing the project.

I deeply express my sincere gratitude to our Project Coordinator Mr. Bhavani Shankar Panda for his valuable suggestions and expert advice.

Last, but not the least, I would like to thank all my team mate for their valuable suggestions and cooperation.

Piyush Kumar(20UG010171)
Anurag Thakur(20UG010169)
Priya Khetan(20UG010184)

ABSTRACT

Rainfall prediction is one of the most important and challenging tasks in the modern world. In general, climate and rainfall are highly non-linear and complicated phenomena, which require advanced computer modelling and simulation for their accurate prediction. To quickly discover and analyze complex patterns and requirements, need the efficient techniques and need to learn from new data which will be necessary for information-intensive applications. One of the solutions for this is Artificial Neural Network. ANN can be used to predict the behaviour of such nonlinear systems. ANN has been successfully used by most of the researchers in this field for the last twenty-five years. The rainfall prediction using ANN technique is more suitable than traditional statistical and numerical methods. Another solution for this is the support vector machine (SVM) is a data-classification algorithm that assigns new data elements to one of the labelled categories, it assumes that the data in question contains some possible target values.

Introduction:

Rainfall prediction is helpful to avoid flood which save lives and properties of humans. Moreover, it helps in managing resources of water. Information of rainfall in prior helps farmers to manage their crops better which result in growth of country's economy. Fluctuation in rainfall timing and its quantity makes rainfall prediction a challenging task for meteorological scientists. In all the services provided by meteorological department, Weather forecasting stands out on top for all the countries across the globe. The task is very complex as it requires numbers of specialized and all calls are made without any certainty.

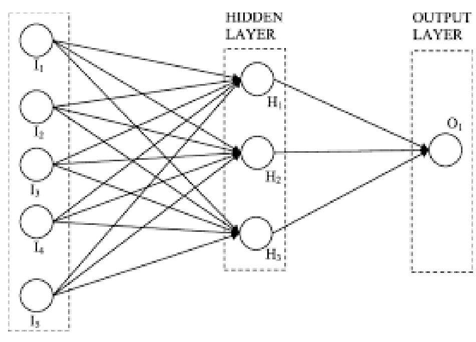
Two widely used methods for rainfall forecasting are: Statistical methods and Numerical Weather Prediction (NWP) model. Nature of rainfall data is non-linear. Frequency, intensity, and amount are main characteristics for time series rainfall. These values can be varied from one position on earth to other position of earth and from one time to other time. Every statistical model has some drawbacks. On a worldwide scale, large numbers of attempts have been made by different researchers to predict rainfall accurately using various techniques. But due to the nonlinear nature of rainfall, prediction accuracy obtained by these techniques is still below the satisfactory level. Artificial neural network algorithm becomes an attractive inductive approach in rainfall prediction owing to their highly nonlinearity, flexibility and data driven learning in building models without any prior knowledge about catchment behavior and flow processes.

Artificial Neural Network:

Using Artificial Neural Networks (ANNs) which are based upon the neural structure of the human brain, complex pattern recognition can be attempted without making any initial assumptions where the data set used is allowed to govern the process by itself. An ANN provides the user a model free tool, which can generate input-output mapping for any set of data, however complex. Training the network with the relevant data enables the network the ability of making predictions based on any input.

In this study, multi-layer perceptron (MLP) is chosen as the network for rainfall predictions. MLP is selected due to its ability for solving complex and non-linear problems.

Backpropagation is used as the learning method. Backpropagation method consists of three steps: feed forward, calculating the error and feed backward. The neural networks model consist of input layer, one hidden layer and output layer. In the first network model, there are n input nodes, x1, x2, ... xn which denotes the rainfall rate of n previous years of the same months of the month that will be predicted. For example, if rainfall rate of Dec 2019 is predicted, the input is the data of rainfall rate of month December in the year of 2018,2017, 2015, 2014, 2013, etc. The number of hidden nodes is set the same as the number of the input nodes. The output is the prediction whether the rainfall will occur in the month or not. The network of the model is illustrated in figure.



Logistics Regression-

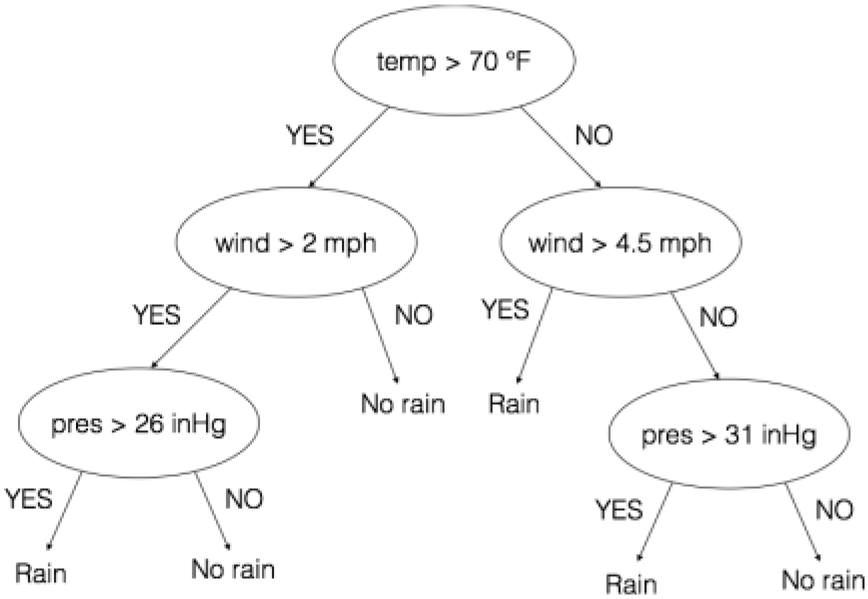
Logistic regression, despite its name, is a classification model rather than regression model. Logistic regression is a simple and more efficient method for binary and linear classification problems. It is a classification model, which is very easy to realize and achieves very good performance with linearly separable classes. It is an extensively employed algorithm for classification in industry. The logistic regression model, like the Adaline and perceptron, is a statistical method for binary classification that can be generalized to multiclass classification. Scikit-learn has a highly optimized version of logistic regression implementation, which supports multiclass classification task (Raschka, 2015). It is used to predict a binary outcome based on a set of independent variables. Logistic regression is the correct type of analysis to use when you are working with binary data. You know you are dealing with binary data when the output or dependent variable is dichotomous or categorical in nature; in other words, if it fits into one of two categories (such as "yes" or "no", "pass" or "fail", and so on).
Logistic regression is another powerful supervised ML algorithm used for binary classification problems (when target is categorical). The best way to think about logistic regression is that it is a linear regression but for classification problems. Logistic regression essentially uses a logistic function defined below to model a binary output variable (Tolles & Meurer, 2016). The primary difference between linear regression and logistic regression is that logistic regression's range is bounded between 0 and 1. In addition, as opposed to linear regression, logistic regression does not require a linear relationship between inputs and output variables. This is due to applying a nonlinear log transformation to the odds ratio.

$$P = \frac{e^{a+bX}}{1+e^{a+bX}}$$

where P is the probability of a 1 (the proportion of 1s, the mean of Y), *e* is the base of the natural logarithm (about 2.718) and *a* and *b* are the parameters of the model. The value of *a* yields P when X is zero, and *b* adjusts how quickly the probability changes with changing X a single unit (we can have standardized and unstandardized b weights in logistic regression, just as in ordinary linear regression). Because the relation between X and P is nonlinear, *b* does not have a straightforward interpretation in this model as it does in ordinary linear regression.

Decision Tree-

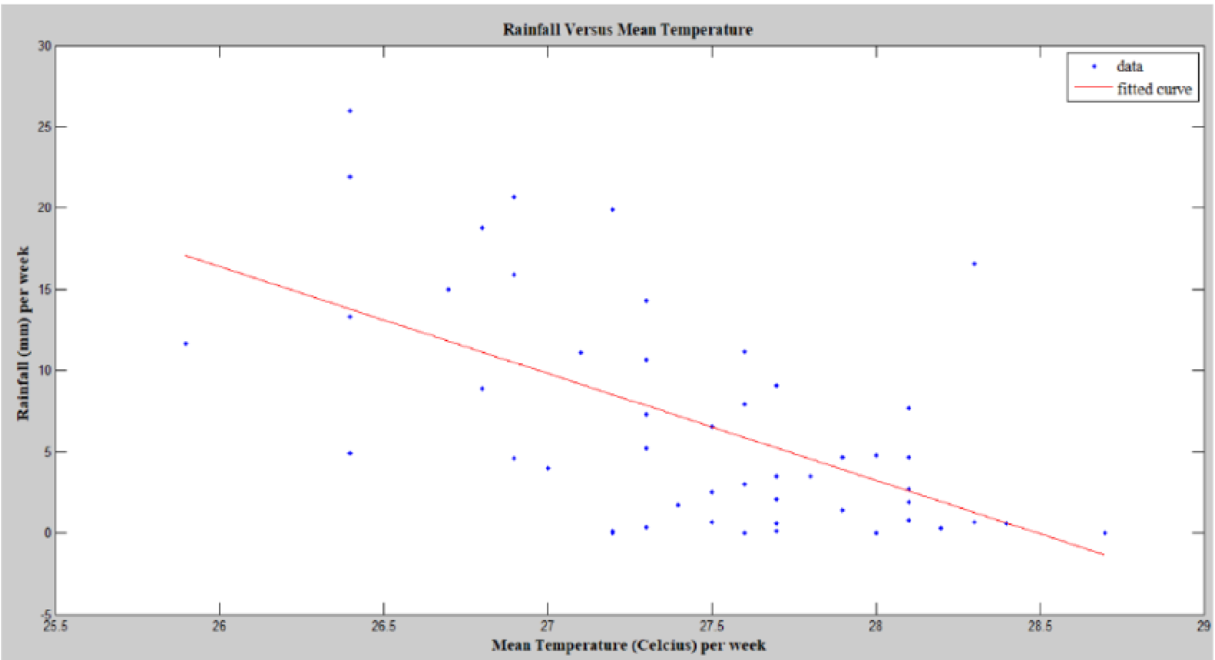
Decision Tree is a machine learning algorithm which you can make the classification and regression. Implementation of this algorithm very easy. In decision tree algorithm, Decision were made by looking at the value which given at the start. The given root separated to the two branches and process goes with these branches.



A tree can be made to learn by splitting the source data set into subsets based on an attribute value test. This process is repeated on each derived subset in a recursive manner called recursive partitioning. The recursion is completed when the subset at a node has all the same value of the target variable or when splitting no longer adds value to the predictions. This process of top-down induction of decision trees is an example of a greedy algorithm, and it is by far the most common strategy for learning decision trees from data in data mining. Decision trees is an example of a greedy algorithm, and it is by far the most common strategy for learning decision trees from data in data mining. In decision tree dependent variable is predicted from the independent variable. Decision tree splits the attributes by using greedy search that optimizes on certain criterion. Test conditions are specified depending on the attributes types whether it is nominal, ordinal or continuous. Determining the best split remains an issue. Greedy method advocates that nodes with homogeneous class distribution are preferred so there is a need to measure a method of node impurity.

Linear Regression-

Linear Regression is a method that describes the relationship between a dependent variable and a set of independent variables. The equation of the line is given as $\boxed{Y=wx+b}$. It provides an estimate of rainfall using various atmospheric variables like cloud cover, humidity, wind, and average temperature to predict rainfall. An estimate of rainfall is easy to determine at any given point since the regression method uses the previous correlation between the various atmospheric variables. The arrangement of neurons forming layers and the connection patterns formed within and between each layer is called Network Structures. ANNs are information processing structures that can solve any problem through learned examples rather than pre-specified algorithms. The proposed system makes use of Linear Regression. In this paper, we have made use of the Chennai dataset. It is given as the input and the file is read by the toolbox. The dataset is then converted into separate data text where the 20 year's plot of rainfall is done. . The dataset is configured by making use of Linear Regression where various parameters are given to initiate the entire network. The prediction values are measured by making use of the toolbox and the graphs are plotted for the obtained predicted values.



ARIMA MODEL-

An ARIMA model is a combination of three operators, i.e., autoregressive (AR) function processed on the historical values of the parameter (rainfall), moving average (MA) function processed on entirely random values, and an integration (I) part to reduce the difference between them.

Purpose:

The purpose of the study is the prediction of the rainfall using historical monthly data based on artificial intelligence methodologies such as support vector machine and artificial neural network. The extraction procedures/algorithms will produce the output by classification of the data according to the categories using SVM and ANN. The similar data will be grouped for the accurate and precise information that will predict rainfall more correctly and with perfect figures. The accurate and exact predictions will help in developing the more appropriate strategies for agriculture and water reserves and will also be informed about the flood to implement precautionary measures. Amongst all weather happenings, rainfall plays the most imperative part in human life. Human civilization to a great extent depends upon its frequency and amount to various scales. Several stochastic models have been attempted to forecast the occurrence of rainfall, to investigate its seasonal variability, to forecast yearly/monthly rainfall over some geographical area. This is the monthly data with all parameters of rainfall including wind speed, direction, air pressure, humidity and temperature. The aim of the proposed study is too effective and efficient in predicting the rainfall with accuracy and precision.

Scope of the Project:

Rainfall prediction is significant not only on the micro but also on the macro level. The study is of significance with respect to its vital contribution in the field of agriculture, water reserve management, flood prediction and management with an intention to ease the people by keeping them updated with the weather and rainfall prediction. It is also important to be utilized by the agricultural industries for keeping their crops safe and ensure the production of seasonal fruits and vegetables by updated rainfall prediction. The study will also be significant for the flood management authorities as more precise and accurate prediction for heavy monsoon rains will keep the authorities alert and focused for an upcoming event that of which the destruction could be minimized by taking precautionary measures. The rainfall prediction will impressively help in dealing with the increasing issue of water resource management; as water is a scarce resource and it needs to get saved for the benefit of human beings themselves. Also, it will help the people to manage and plan their social activities accordingly.

Objective of the Study:

Accuracy of rainfall forecasting has great importance for countries like India whose economy is largely dependent on agriculture. Thus, the objective of this survey is to make the prediction of rainfall more accurate in the recent future and also to predict the rainfall values for nonlinear data.

Proposed System:

A detailed survey on rainfall predictions using Artificial Neural Network architecture over twenty-five years is done. From the survey it has been found that most of the researchers used different models for rainfall prediction, but keras model of ANN gives significant results. ANN is the model with least mean squared error and accurate prediction. The survey also gives a conclusion that the forecasting techniques like SVC, SVR of SVM are suitable to predict rainfall than other forecasting techniques such as statistical and numerical methods. However, some limitation of those methods has been found. The extensive references in support of the different developments of ANN research provided should be of great help to ANN researchers to accurately predict rainfall in the future.

Advantages ·

>High prediction accuracy.

>Hold perfectly good for large scale datasets with large number of variables.

>Integral variable selection based on importance and variable interaction.

>Deals efficiently with data having missing values.

>Computation of relation between variables and classification. Proximity calculation between cases.

>Can be used for unsupervised learning and outlier detection.

>Internal unbiased estimation of the generalization error.

Sequence

Initially the data is collected and it is pre-processed. After pre-processing is completed, we sample the data. After sampling training data sets are obtained. Then the training data are pre-processed and fetched into the algorithm. The output of the learning algorithm is analyzed and hence the forecast is predicted. Both ANN and SVM are using the same architecture to predict the rainfall. The algorithm inputs the date for which the rainfall needs to be predicted. After the successful intake of the input the program is trained with the rainfall data and will predict whether rainfall will occur or not for that input date.

Methodology ·

- Data Analysis – Data analysis is a process of inspecting, cleansing, transforming, and modelling data with the goal of discovering useful information, informing conclusions, and supporting decision-making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and is used in different business, science, and social science domains. In today's business world, data analysis plays a role in making decisions more scientific and helping businesses operate more effectively. Analysis of the rainfall data and affects data and plotting several graphs for further processing.

- Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System),[3] and its primary author and maintainer is François Chollet, a Google engineer. Chollet also is the author of the XCeption deep neural network model

- Keras contains numerous implementations of commonly used neural network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier. The code is hosted on GitHub, and community support forums include the GitHub issues page, and a Slack channel. In addition to standard neural networks, Keras has support for convolutional and recurrent neural.networks. It supports other common utility layers like dropout, batch normalization, and pooling.

- Regression - In statistical modeling, regression analysis is a set of statistical processes for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables (or 'predictors'). More specifically, regression analysis helps one understand how the typical value of the dependent variable (or 'criterion variable') changes when any one of the independent variables is varied, while the other independent variables are held fixed.

- Most commonly, regression analysis estimates the conditional expectation of the dependent variable given the independent variables – that is, the average value of the dependent variable when the independent variables are fixed. Less commonly, the focus is on a quantile, or other location parameter of the conditional distribution of the dependent variable given the independent variables. In all cases, a function of the independent variables called the regression function is to be estimated. In regression analysis, it is also of interest to characterize the variation of the dependent variable around the prediction of the regression function using a probability distribution. A related but distinct approach is Necessary Condition Analysis[1] (NCA), which estimates the maximum (rather than average) value of the dependent variable for a given value of the independent variable (ceiling line rather than central line) in order to identify what value of the independent variable is necessary but not sufficient for a given value of the dependent variable. [

- Regression analysis is widely used for prediction and forecasting, where its use

has substantial overlap with the field of machine learning. Regression analysis is also used to understand which among the independent variables are related to the dependent variable, and to explore the forms of these relationships. In restricted circumstances, regression analysis can be used to infer causal relati onships between the independent and dependent variables. However this can lead to illusions or false relationships, so caution is advisable.

Code-

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

```
!pip install matplotlib
```

```
df = pd.read_csv('weatherAUS.csv')
df.head()
```

```
df.describe()
```

```
df.shape
```

```
df =
df.drop(["Evaporation","Sunshine","Cloud9am","Cloud3pm","Location",
"Date"], axis =1)
df.head()
```

```
df = df.dropna(axis = 0)
df.shape
```

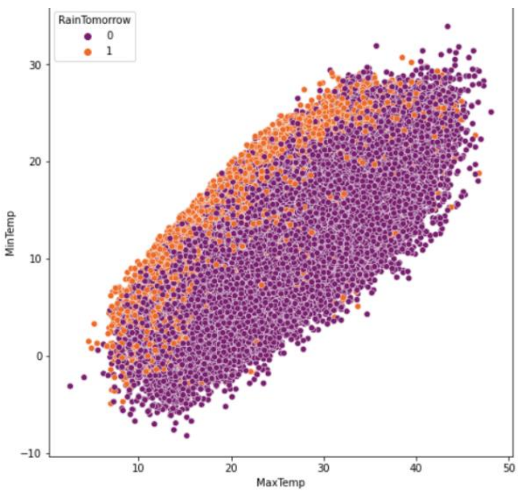
```
df.columns
```

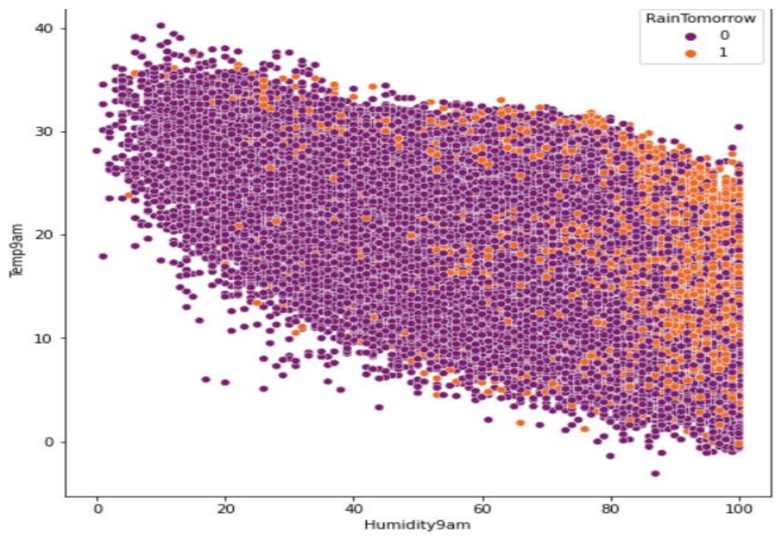
```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['WindGustDir'] = le.fit_transform(df['WindGustDir'])
df['WindDir9am'] = le.fit_transform(df['WindDir9am'])
df['WindDir3pm'] = le.fit_transform(df['WindDir3pm'])
df['RainToday'] = le.fit_transform(df['RainToday'])
df['RainTomorrow'] = le.fit_transform(df['RainTomorrow'])
```

```
x = df.drop(['RainTomorrow'], axis = 1)
y = df['RainTomorrow']
```

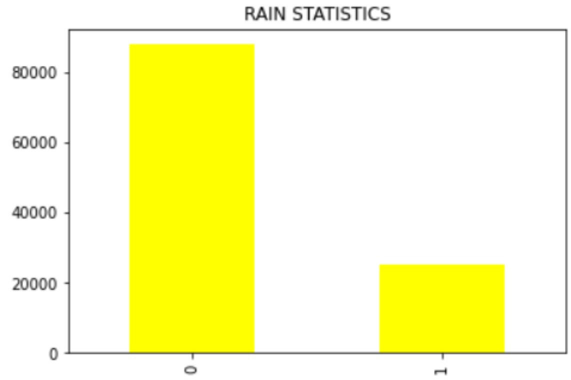
```
x.head()
```

```
plt.figure(figsize = (8,8))
sns.scatterplot(x = 'MaxTemp', y = 'MinTemp', hue = 'RainTomorrow' ,
palette = 'inferno',data = df)
plt.figure(figsize = (8,8))
sns.scatterplot(x = 'Humidity9am', y = 'Temp9am', hue = 'RainTomorrow'
, palette = 'inferno',data = df)
```

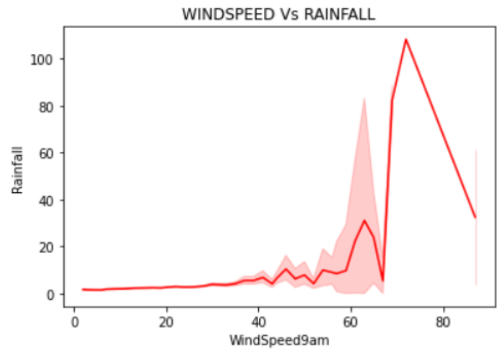




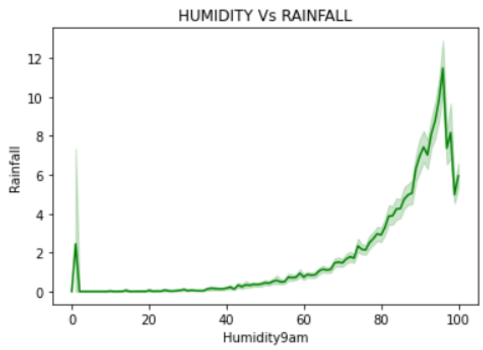
```
df['RainTomorrow'].value_counts().plot(kind='bar',color="yellow",title=
"RAIN STATISTICS")
```



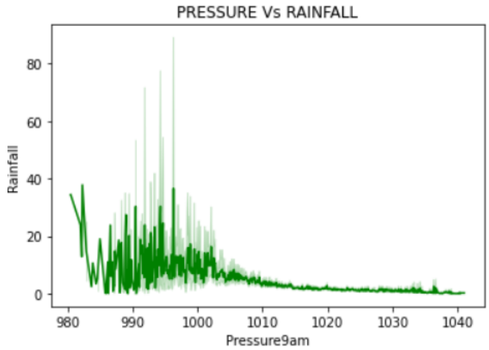

```
plt.title("WINDSPEED Vs RAINFALL")
sns.lineplot(data=df,x='WindSpeed9am',y='Rainfall',color='red')
```



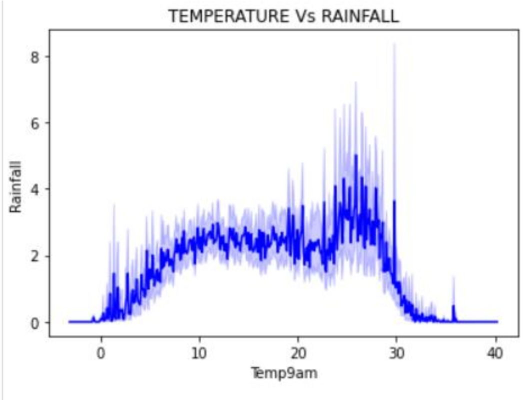

```
plt.title("HUMIDITY Vs RAINFALL")
sns.lineplot(data=df,x='Humidity9am',y='Rainfall',color='green')
```



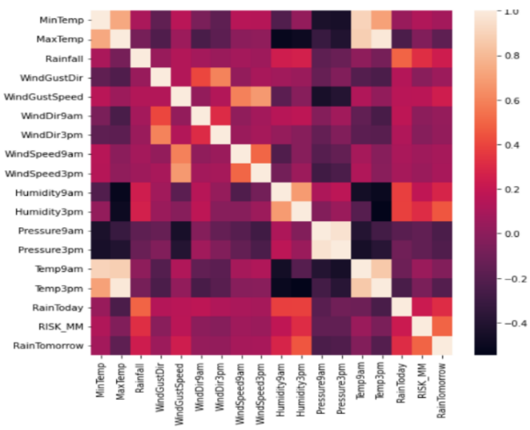

```
plt.title("PRESSURE Vs RAINFALL")
sns.lineplot(data=df,x='Pressure9am',y='Rainfall',color='green')
```



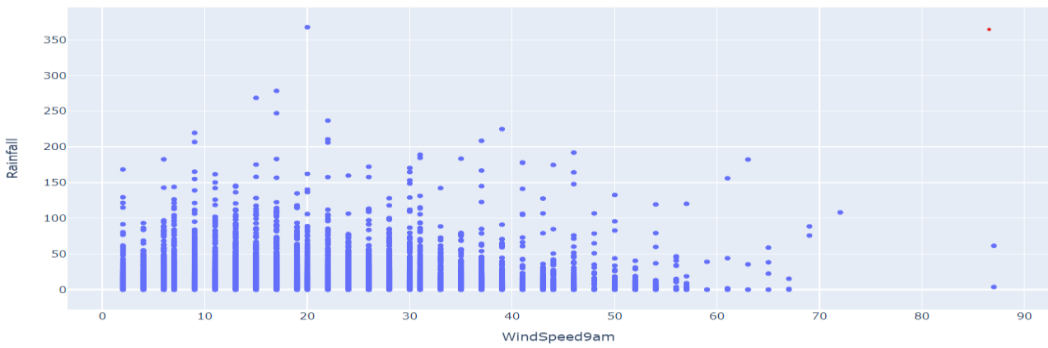

```
plt.title("TEMPERATURE Vs RAINFALL")
sns.lineplot(data=df,x='Temp9am',y='Rainfall',color='blue')
```



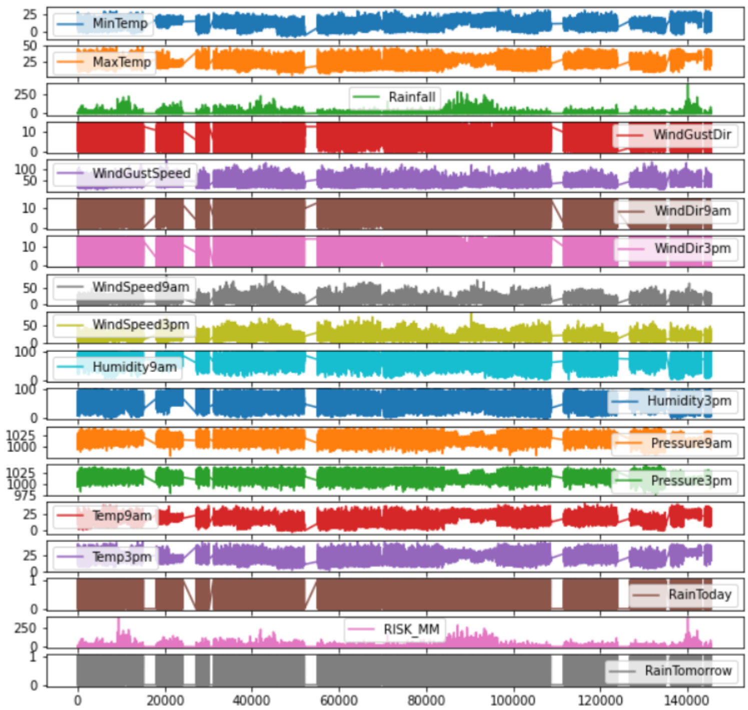

```
plt.figure(figsize = (8,8))
sns.heatmap(df.corr())
```




```
import plotly.express as px
graph = px.scatter(df,x='WindSpeed9am',y='Rainfall')
graph.show()
```



```
df.plot(subplots=True,figsize=(10,10))
```




```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size =
0.2)
```

```
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score
```

```
#logistic regression
from sklearn.linear_model import LogisticRegression
import time
t0 = time.time()
lr = LogisticRegression()
lr.fit(x_train,y_train)
predictions = lr.predict(x_test)
tt_lr=int(time.time())
print(confusion_matrix(y_test, predictions))
print(classification_report(y_test, predictions))
accuracy_lr=accuracy_score(y_test, predictions)
print("THE ACCURACY USING LOGISTIC REGRESSION: ",accuracy_lr)
print("THE TIME TAKEN USING LOGISTIC REGRESSION: ",tt_lr)
```

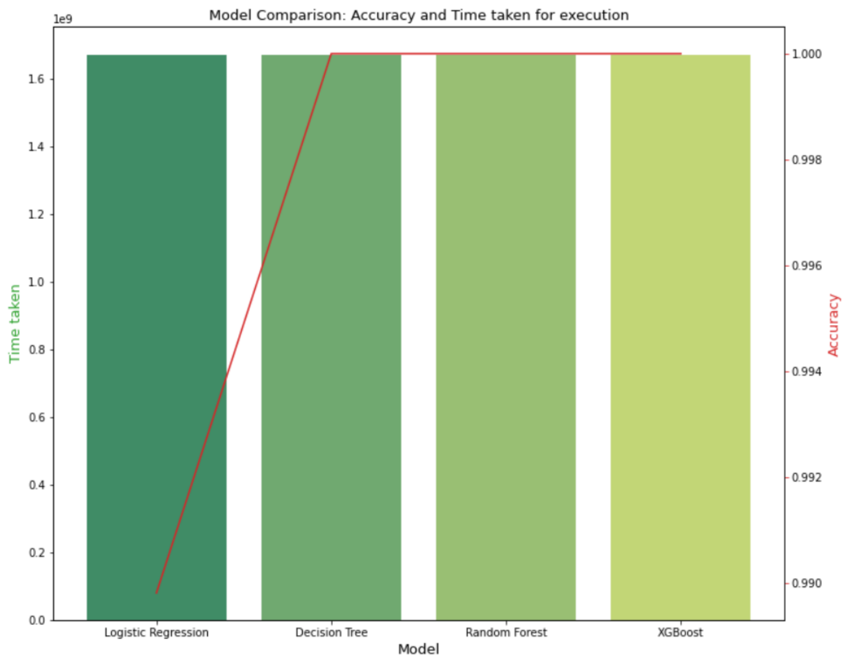
```
#decision tree
from sklearn.tree import DecisionTreeClassifier
t0 = time.time()
dt = DecisionTreeClassifier()
dt.fit(x_train,y_train)
predictions = dt.predict(x_test)
tt_dt=int(time.time())
print(confusion_matrix(y_test, predictions))
print(classification_report(y_test, predictions))
accuracy_dt=accuracy_score(y_test, predictions)
print("THE ACCURACY USING DECISION TREE: ",accuracy_dt)
print("THE TIME TAKEN USING DECISION TREE: ",tt_dt)
```

```
#random forest classifier
from sklearn.ensemble import RandomForestClassifier
t0=time.time()
rf = RandomForestClassifier()
rf.fit(x_train,y_train)
predictions = rf.predict(x_test)
tt_rf=int(time.time())
print(confusion_matrix(y_test, predictions))
print(classification_report(y_test, predictions))
accuracy_rf=accuracy_score(y_test, predictions)
print("THE ACCURACY USING RANDOM FOREST CLASSIFIER: ",accuracy_rf)
print("THE TIME TAKEN USING RANDOM FOREST CLASSIFIER: ",tt_rf)
```

```
import xgboost as xgb
t0=time.time()
xgb = xgb.XGBClassifier()
xgb.fit(x_train, y_train)
pred = xgb.predict(x_test)
tt_xgb=int(time.time())
acc_xgb=accuracy_score(y_test,pred)

print('f1',classification_report(y_test,pred))
print('CONFUSION MATRIX',confusion_matrix(y_test,pred))

print('ACCURACY USING XG BOOST: ',acc_xgb)
print('TIME TAKEN USING XG BOOST: ',tt_xgb)
```

```
accuracy_scores = [accuracy_lr, accuracy_dt, accuracy_rf,acc_xgb]
tt = [tt_lr, tt_dt, tt_rf,tt_xgb]

model_data = {'Model': ['Logistic Regression','Decision Tree','Random 
Forest','XGBoost'],
              'Accuracy': accuracy_scores,
              'Time taken': tt}
```

```
data = pd.DataFrame(model_data)

fig, ax1 = plt.subplots(figsize=(12,10))
ax1.set_title('Model Comparison: Accuracy and Time taken for
execution', fontsize=13)
color = 'tab:green'
ax1.set_xlabel('Model', fontsize=13)
ax1.set_ylabel('Time taken', fontsize=13, color=color)
ax2 = sns.barplot(x='Model', y='Time taken', data = data,
palette='summer')
ax1.tick_params(axis='y')
ax2 = ax1.twinx()
color = 'tab:red'
ax2.set_ylabel('Accuracy', fontsize=13, color=color)
ax2 = sns.lineplot(x='Model', y='Accuracy', data = data, sort=False,
color=color)
ax2.tick_params(axis='y', color=color)
```




```
print("The accuracy of Logistic Regression is:
{0}".format(accuracy_lr))
print("The time taken for Logistic Regression is: {0}".format(tt_lr))
print("\n\nThe accuracy of Decision Tree is: {0}".format(accuracy_dt))
print("The time taken for Decision Tree is: {0}".format(tt_dt))
print("\n\nThe accuracy of Random forest classifier is:
{0}".format(accuracy_rf))
print("The time taken for Random forest classifier is:
{0}".format(tt_rf))
print("\n\nThe accuracy of XGBoost is:",(acc_xgb))
print("The time taken for XGBoost is:",(tt_xgb))
```

```
X = df[['MaxTemp', 'MinTemp','Rainfall','WindSpeed9am','WindSpeed3pm']]
y = df['RainTomorrow']
```

```
from sklearn import linear_model
regr = linear_model.LinearRegression()
regr.fit(X, y)
maxi=float(input("ENTER MAX TEMP:"))
mini=float(input("ENTER MIN TEMP:"))
rainf=float(input("ENTER RAINFALL:"))
wind9=float(input("ENTER WINDSPEED AT 9 AM:"))
wind3=float(input("ENTER WINDSPEED AT 3 PM:"))
#hum9=float(input("ENTER HUMIDITY AT 9 AM:"))
#hum3=float(input("ENTER HUMIDITY AT 3 PM:"))
#temp9=float(input("ENTER TEMPERATURE AT 9 AM:"))
#temp3=float(input("ENTER TEMPERATURE AT 3 PM:"))

y=df.RainTomorrow.values.reshape(-1,1)
raintom =regr.predict([[maxi,mini,rainf,wind9,wind3]])
if(raintom<0.5):
    print("RAIN TOMORROW: NO")
if(raintom>=0.5 ):
    print("RAIN TOMORROW: YES")
print(raintom)
```

```
X = df[['MaxTemp', 'MinTemp','Rainfall','WindSpeed9am','WindSpeed3pm']]
y = df['RainToday']
from sklearn import linear_model
regr = linear_model.LinearRegression()
regr.fit(X, y)
maxi=float(input("ENTER MAX TEMP:"))
mini=float(input("ENTER MIN TEMP:"))
rainf=float(input("ENTER RAINFALL:"))
wind9=float(input("ENTER WINDSPEED AT 9 AM:"))
wind3=float(input("ENTER WINDSPEED AT 3 PM:"))
#hum9=float(input("ENTER HUMIDITY AT 9 AM:"))
#hum3=float(input("ENTER HUMIDITY AT 3 PM:"))
#temp9=float(input("ENTER TEMPERATURE AT 9 AM:"))
#temp3=float(input("ENTER TEMPERATURE AT 3 PM:"))

y=df.RainTomorrow.values.reshape(-1,1)
raintod =regr.predict([[maxi,mini,rainf,wind9,wind3]])
if(raintod<0.5):
    print("RAIN TODAY: NO")
if(raintod>=0.5 ):
    print("RAIN TODAY: YES")
print(raintod)
```

```
X = df[['MaxTemp', 'MinTemp','WindSpeed9am','WindSpeed3pm']]
y = df['Rainfall']
from sklearn import linear_model
regr = linear_model.LinearRegression()
regr.fit(X, y)
maxi=float(input("ENTER MAX TEMP:"))
mini=float(input("ENTER MIN TEMP:"))
wind9=float(input("ENTER WINDSPEED AT 9 AM:"))
wind3=float(input("ENTER WINDSPEED AT 3 PM:"))
#hum9=float(input("ENTER HUMIDITY AT 9 AM:"))
#hum3=float(input("ENTER HUMIDITY AT 3 PM:"))
#temp9=float(input("ENTER TEMPERATURE AT 9 AM:"))
#temp3=float(input("ENTER TEMPERATURE AT 3 PM:"))

y=df.Rainfall.values.reshape(-1,1)
preci=regr.predict([[maxi,mini,wind9,wind3]])
print("THE PRECIPITATION LEVEL IS: ",preci)
```

```
if preci<15:
    print("LESS RAINFALL AND NO ALERT")
if preci>=15 and preci<65:
    print("MODERATE RAINFALL AND GREEN ALERT")
if preci>=65 and preci<115:
    print("HEAVY RAINFALL AND YELLOW ALERT")
if preci>=115 and preci<205:
    print("VERY HEAVY RAINFALL AND ORANGE ALERT")
if preci>=205 :
    print("EXTREMELY HEAVY RAINFALL AND RED ALERT")
```

```
from sklearn.model_selection import train_test_split
```

```
x = df.iloc[:,df.columns!='RainTomorrow'] #atributes / data
y = df.iloc[:,df.columns=='RainTomorrow'] #outcomes / label
```

```
from sklearn.ensemble import RandomForestClassifier
```

```
#divide 20% for test data & 80% for train data
xtrain, xtest, ytrain, ytest = train_test_split(x,y,test_size=0.2)
```

```
#call 'RandomForestClassifier' algorithm
model=RandomForestClassifier()
```

```
#fit the training data into the algo
model.fit(xtrain,ytrain.values.ravel())
```

```
# pass the test data into the machine to see the result
predict_output = model.predict(xtest)
```

```
# test data result (outcomes) 1- RAIN TOMORROW 0- NO RAIN
print(predict_output[0:1000])
```

```
import requests
city = input('INPUT CITY NAME: ')
print('Displaying Weather report for: ' + city)
url = 'https://wttr.in/{}'.format(city)
res = requests.get(url)

print(res.text)
```

OUTPUT-

If I enter city name as **“GUNUPUR”**, it will show result like this—

```
INPUT CITY NAME: Gunupur
Displaying Weather report for: Gunupur
Weather report: Gunupur

   \  /       Partly cloudy
 _ /"".-.     +78(84) °F
   \_(   ).   ↖ 6 mph
   /(___(__)  6 mi
              0.0 in
                                                       ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤  Wed 23 Nov ├───────────────────────┬──────────────────────────────┐
│            Morning           │             Noon      └──────┬──────┘     Evening           │             Night            │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────┼──────────────────────────────┤
│    \   /     Sunny           │             Cloudy           │   \  /       Partly cloudy   │   \  /       Partly cloudy   │
│     .-.      +78(80) °F      │      .--.   +84(86) °F       │ _ /"".-.     +75(78) °F      │ _ /"".-.     +73(77) °F      │
│  ― (   ) ―   ↓ 1-2 mph       │   .-(    ).  ↖ 2-3 mph       │   \_(   ).   ↖ 3-4 mph       │   \_(   ).   ↓ 0-1 mph       │
│     `-’      6 mi            │  (___.__)__) 6 mi            │   /(___(__)  6 mi            │   /(___(__)  6 mi            │
│    /   \     0.0 in | 0%     │              0.0 in | 0%     │              0.0 in | 0%     │              0.0 in | 0%     │
└──────────────────────────────┴──────────────────────────────┴──────────────────────────────┴──────────────────────────────┘
                                                       ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤  Thu 24 Nov ├───────────────────────┬──────────────────────────────┐
│            Morning           │             Noon      └──────┬──────┘     Evening           │             Night            │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────┼──────────────────────────────┤
│              Cloudy          │ _`/"".-.     Patchy rain po…│   \  /       Partly cloudy   │   \  /       Partly cloudy   │
│      .--.    +78(80) °F      │  ,\_(   ).   +80(84) °F      │ _ /"".-.     +75(78) °F      │ _ /"".-.     +71(77) °F      │
│   .-(    ).  ↙ 0-1 mph       │   /(___(__)  ← 1-2 mph       │   \_(   ).   ↖ 3-4 mph       │   \_(   ).   ↙ 1-2 mph       │
│  (___.__)__) 6 mi            │     ‘ ‘ ‘ ‘  6 mi            │   /(___(__)  6 mi            │   /(___(__)  6 mi            │
│              0.0 in | 0%     │    ‘ ‘ ‘ ‘   0.0 in | 66%    │              0.0 in | 0%     │              0.0 in | 0%     │
└──────────────────────────────┴──────────────────────────────┴──────────────────────────────┴──────────────────────────────┘
                                                       ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤  Fri 25 Nov ├───────────────────────┬──────────────────────────────┐
│            Morning           │             Noon      └──────┬──────┘     Evening           │             Night            │
├──────────────────────────────┼──────────────────────────────┼──────────────────────────────┼──────────────────────────────┤
│    \   /     Sunny           │    \   /     Sunny           │    \   /     Clear           │    \   /     Clear           │
│     .-.      +77(80) °F      │     .-.      +84(86) °F      │     .-.      +73(77) °F      │     .-.      69 °F           │
```

<u>CONCLUSION-</u>

Rainfall is one the most significant natural phenomenon that is not only important for the human beings only but the living beings. Due to the changing climatic conditions, rainfall cycles are also changing and the temperature of the earth is rising. The changing temperature is also affecting the agriculture, industry and sometimes may cause flooding and land slide. Therefore, it is essential for the human beings to keep a check upon this natural phenomenon in order to survive. The water is a scarce natural resource without which human life is impossible and also there is no substitute to this natural resource. Thus, predicting the rainfall for agriculture and water reserves, also it also good for keeping human beings alert of natural disasters like flood and landslide. However, to overcome these issues and meet the demands, a system to forecast rainfall is essential using artificial intelligence of neural that is popular within the modern technology.

The study aimed at building a predicting system using neural networks that could predict monthly rainfall accurately and efficiently with minimum error. The study incorporated different areas and used their rainfall data with different neural networks like ANFIS and ANN, through training the networks with these inputs and outputs. The trained data is tested and then validated by making a comparison between actual and predicted data. The system used feature extraction to deduce the output prediction that could be more precise and accurate. The neural networks with different algorithms and functions were trained with rainfall parameters and the previous rainfall data to predict the results in this study. After training and testing; the results were compared to check the efficiency of the system; the RMSE's were recorded to make sure that the system will operate not only to make the prediction but also the accurate data will be obtained. The study utilized back propagation, NARX and Hybrid algorithms to forecast the rainfall.

Lastly, 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 using ANFIS only. The AFIS model outlined efficiency for Erbil, Nicosia and Famagusta. The NARX network performed not so well in the comparison for actual and predicted data.