CSE3505 – Foundations of Data Analytics

Project Report

Weather Analysis of Major Cities

By

20BCE1848	Altaf Pathan
20BCE1287	Kumar Siddharth
20BCE1196	Prakhar Sachan
20BCE1098	Vishvajeet Ravalekar

B. Tech Computer Science and Engineering

Submitted to

Trilok Nath Pandey

School of Computer Science and Engineering

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

November 2022

<u>DECLARATION</u>

<Times New Roman, Font 14, Underlined, Bold, CAPS>

I hereby declare that the report titled "Weather Analysis of Major Cities" submitted by me to VIT Chennai is a record of bona-fide work undertaken by me under the supervision of **Trilok Nath Pandey**, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai.

Signature of the Candidate

(To be (digital)signed by the student)

(Altaf Pathan)

Reg. No. 20BCE1848

CERTIFICATE

Certified that this project report entitled "Weather Analysis of Major Cities" is a bonafide work of Altaf Pathan **(20BCE1848), Vishvajeet Ravalekar (20BCE 1098) Kumar Siddharth (20BCE1287) , Prakhar Sachan (20BCE1196)** and they carried out the Project work under my supervision and guidance for CSE3505 – Foundations of Data Analytics.

Trilok Nath Pandey

SCOPE, VIT Chennai

ACKNOWLEDGEMENT

We would like to convey our sincere gratitude to our project advisor, Dr.Trilok Nath Pandey, SCOPE, for his constant support and insightful advice that he provided to us in a pleasant manner during the project work as well as for inspiring us to complete the study on the topic “Weather Analysis of Major Cities ”. We would want to take this opportunity to thank the faculties of the college for their guidance and support throughout the project. Lastly, we wish to express our gratitude to our parents, relatives, and friends for their support throughout the execution and helped us in every possible way and displayed appreciation for our project and for the opportunity they provided us in undergoing this course at such a prominent institution.

(Altaf Pathan)

Reg. No. 20BCE1848

ABSTRACT

Many cities in the world have a typical weather condition consisting of various seasons and geographical conditions. They have extreme high temperatures in regions like Arizona, Sahara, Thar and cold climate at Himalayas, Tundra regions and heavy rainfall at Chirapunji, Japan. These extreme variations in temperatures make us to feel difficult in inferring / predictions of weather effectively. It requires higher scientific techniques / methods for plotting and pre-processing of data. In this project, we will apply different methods and algorithms for grouping similar data sets together and also apply classification technique along with linear regression analysis. Programming will be done in programming languages like R and Python.

CONTENTS

<Times New Roman, Font 14, Underlined, Bold, CAPS>

<Times New Roman, Font 12>

1. Introduction

This paper gives a brief summary of weather forecast trends, challenges, and the nature of their occurrence, as well as existing and promising solutions. The neural network architecture is offered as a possible method for improving the accuracy of weather forecasts produced by various regional models. This design enables for the prediction of atmospheric model forecast errors as well as their subsequent corrections. Experiments with various histories of regional model errors are performed. It is demonstrated that the proposed architecture allows for the improvement of a weather forecast. The initial attempt at weather prediction necessitated a larger workforce. Weather prediction has returned to the early models in terms of similarity, thanks to the development of powerful and better modeling tools. The forecast equations in Weather Prediction are then simple-basic equations. Because equations dictate how meteorological variables change over time, if the initial state of the atmosphere is known, equations may be used in our project to anticipate new values for those variables in the future. Weather data is gathered for a short-range over a specific region at a specific station. The results suggest that it can more precisely and reliably forecast meteorological conditions. In the current era of machine learning and technologies, weather prediction is rapidly gaining ubiquity. Predicting the climate for an extended period of time is critical. Decision trees, K-NN, and Random Forest etc. algorithms are valuable assets that have been used in a variety of prediction applications, such as flood prediction and storm detection. The decision tree, K-NN, and random forest method calculations are used in this research to present a simple technique for weather prediction of future years by employing historical data analysis, with the best accuracy result of these three algorithms. Weather prediction is important in everyday life, and the forecast in this study is based on rain variations in a specific area. All of these methods compute mean values, medians, confidence intervals, and probability, as well as plot the differences between the three algorithms' plots, and so on. Finally, utilizing the algorithms used in this study, we can forecast whether it will rain or not. The dataset is totally dependent on the weather of a given area including a few objects like a year, month, forecasted values, and so on. Weather changes have a huge negative impact on the ecosystem and might suddenly precipitate natural disasters. There are numerous machine learning approaches and algorithms that can be used to forecast these changes and predict them early. It has been noted that, based on past research, there are a variety of additional ways to weather prediction. Various parameters such as rain, temperature, humidity, wind direction, precipitation, evaporation, and so on are taken into account based on these

What is the Weather Forecast?

Weather forecasting is the use of science and technology to forecast atmospheric conditions for a specific location and time. For millennia, people have sought to predict the weather informally, and systematically since the nineteenth century. Weather predictions are created by gathering quantitative data on the current state of the atmosphere, land, and ocean, and then applying meteorology to project how the atmosphere will change at a certain location. Weather forecasting is currently based on computer-based models that take many atmospheric aspects into account, rather than being estimated manually based on changes in barometric pressure, present weather conditions, and sky condition or cloud cover. Pattern recognition skills, teleconnections, knowledge of model performance, and understanding of model biases are still required for selecting the best potential forecast model on which to base the forecast. The chaotic nature of the atmosphere, the massive computational power required to solve the equations that describe the atmosphere, land, and ocean, the error involved in measuring initial conditions, and an incomplete understanding of atmospheric and related processes all contribute to forecasting's inaccuracy. As a result, as the time difference between now and the time for which the forecast is being produced grows, projections become less accurate. The usage of ensembles and model consensus can help to reduce the error and increase the forecast's confidence level. Weather forecasts have a wide range of applications. Weather warnings are crucial forecasts because they safeguard people and property. Agricultural forecasts based on temperature and precipitation are critical, and traders in commodities markets rely on them. Many people use weather forecasts to decide what to wear on a given day on a daily basis. Because heavy rain, snow, and wind chill significantly limit outdoor activities, forecasts can be used to schedule activities around these phenomena, as well as to prepare ahead and survive them.

What is the use of weather prediction?

Weather prediction's purpose is to give information that people and organisations can use to reduce weather-related losses and improve societal advantages, such as life and property protection, public health and safety, and economic prosperity and quality of life.

1.1 Objective and goal of the project

The objective and goal of the project is to predict the rainfall in a particular area after analysis the data of the previous day like temperature, pressure, wind speed, temperature range, etc.

1.2 Problem Statement

To analyze the weather of previous day and predict the weather of the next day.

1.3 Motivation

Chennai’s weather is very unpredictable. Most of the locals of Chennai city often forgets to carry umbrella’s or rain coats with them. To make them aware about the future, we chose this problem statement.

1.4 Challenges

The challenge was to find a proper dataset which included all the parameters which was needed to predict the output. Another challenge was figuring out proper machine learning algorithm to predict the accuracy.

2. Literature Survey

On a worldwide scale, large numbers of attempts have been made by different researchers to forecast Weather accurately using various techniques. But due to the nonlinear nature of Weather, prediction accuracy obtained by these techniques is still below the satisfactory level. Ali, Lin etc [1] developed an ANN technique to estimate tropical cyclone heat potential (TCHP) for estimating the Cyclone and Intensity prediction. They estimated TCHP by 1) an ANN technique, 2) a two-layer reduced gravity model, and 3) a multiple regression technique and compared the estimations with the in situ observations. Out of the three methods, they found that ANN approach has given the best results. The results suggest the utility of the ANN technique in estimating TCHP with better accuracy in the North Indian Ocean that certainly, in turn, helps in improving the cyclone track and intensity predictions. Sarma, Konwar, Das etc [2] showed that Artificial Neural Network (ANN) can also be combined with different methods. A neural network model for rainfall retrieval over ocean from remotely sensed microwave (MW) brightness temperature (BT) is described. They Proposed, a soft computing approach for rainfall prate estimation over ocean using online feature selection, clustering, and hybrid neural network. In this study, they applied an online feature selection (FS) algorithm to the BT dataset obtained from TMI. The nine-channel BT data are the input feature to this feature selection algorithm. It selects the most relevant channels both vertical as well as horizontal. A k-means clustering algorithm is then applied to the dataset of selected features. Separate multilayer perceptron (MLP) neural networks are trained for each of the clustered data. These trained MLPs are then combined to form a hybrid network. The results showed that hybrid network ANN-Hyb provided better instantaneous rain fall rate estimation compared to ANN alone. Chaudhuri & Chattopadhyay [3] designed a Feed forward multi-layered artificial neural network model to estimate the maximum surface temperature and relative humidity needed for the genesis of severe thunderstorms over Calcutta. Prediction error is computed and compared for single layer network and one hidden layer artificial neural network. Result reveals the efficiency of the one hidden layer ANN. The performance of the model is found to be good and showed that the neural network technique is of great use in forecasting the occurrence of high frequency small-scale weather systems. The forecasting from single-layer-neural net is compared with the forecasting from the second order autoregressive process. It is found that forecasting yield is better in case of single-layerneural net than in it. From this, it can be concluded that one hidden-layer neural network is an efficient forecasting tool by which an estimation of maximum surface temperature and maximum relative humidity can be obtained. This estimation can help in predicting a probable thunderstorm day with one day or 24 hrs in advance. Another researcher F. Mekanik and M. A. Imteaz [4] found that Australian rainfall is also affected by these key modes of complex climate variables. On the other hand, few attempts have been made to establish the combined effect of these indices on rainfall in order to develop a better understanding and forecasting system. Since rainfall is a complicated atmospheric phenomenon, linear techniques might not be sufficient enough to capture its characteristics. This research attempts to find a nonlinear relationship between the Victorian rainfall and the lagged-indices affecting the region using Artificial Neural

Networks (ANN). It was discovered that ANN modeling is able to provide higher correlations using the laggedindices to forecast spring rainfall in compared to linear methods. Using these indices in an ANN model increased the model correlation up to 99%, 98% and 43% for the three case study stations of Horsham, Melbourne and Orbost in Victoria, Australia respectively. Luk, Ball and Sharma [5] described that due to the complexity of the atmospheric processes by which rainfall is generated and the lack of available data on the necessary temporal and spatial scales, it is not feasible generally to forecast rainfall using a physically based process model. They presented the results of a study investigating the application of ANN to forecast the spatial distribution of rainfall for an urban catchment. Three types of ANNs suitable for this task were identified, developed, and compared; these networks were (i) multilayer feed forward neural network (MLFN),(ii) Elman partial recurrent neural network (Elman), and (iii) time delay neural network (TDNN).All the above alternative networks could make reasonable forecast of rainfall one time step (15 minutes) ahead for 16 gauges concurrently. In addition, the following points were observed. (a)For each type of network, there existed an optimal complexity, which was a function of the number of hidden nodes and the lag of the network. (b)All three networks had comparable performance when they were developed and trained to reach their optimal complexities. (c)Networks with lower lag tended to outperform the ones with higher lag. This indicates that the 15 min. rainfall time series have very short term memory characteristics. Another researcher [6] described a weather forecasting problem-rain fall using different neural network architectures namely Electronic Neural Network (ENN) model and optoelectronic neural network model. They experimented using these two models and the percentage of correctness of the rainfall estimation of the neural network models and the meteorological experts are compared. The results of the ENN are compared with the results of the opto-electronic neural network for the estimation of rainfall. The accuracy of the results, obtained using ENN and opto-electronic neural network models, is compared with two metrological experts. The performance of optoelectronic neural network, which is better than the performance of ENN, is reported. Hayati and Mohebi [7] shows that how ANN can be used for Forecasting Weather for the city of Iran. They utilizes ANN for one day ahead prediction of weather parameter i.e. temperature of city of Iran. Their study was based on most common neural network model Multilayer Perceptron (MLP) which is trained and tested using ten years past metrological data. In order to improve the accuracy of prediction, they split data into four season's i.e. spring, summer, fall and winter and then for each season one network is presented. They showed that MLP network with this structure has minimum error between exact and predicted values at each day and has a good performance, reasonable prediction accuracy and minimum prediction error in general. The forecasting reliability was evaluated by computing the mean absolute error between the exact and predicted values. The result shows that this network can be an important tool for temperature forecasting. Kaur and Singh [8] in his work uses Multi-Layer Perceptron (MLP) to model forecasting system and used Back Propagation algorithm to train the network. The network is trained and tested with actual data of the past ten years which comes from meteorological department. The results show that minimum temperature can be predicted with reasonable accuracy by using the Artificial Neural Network. Thus from this work they concluded that a feed-forward NN model using back-propagation algorithm is developed to identify the minimum temperature. The results show that an appropriate accuracy can be achieved using this network. Further, this approach is also able to determine the values of other parameters like maximum relative humidity, minimum relative humidity, maximum temperature etc in a particular year. Artificial neural networks have been extensively used in these days in various aspects of science and engineering because of its ability to model both linear and non-linear systems without the need to make assumptions as are implicit in most traditional statistical approaches. Vamsidhar, Rao, satapati etc[9] used the back propagation neural network model for predicting the rainfall based on humidity, dew point and pressure in the country INDIA. Two-Third of the data was used for training and One-third for testing. The number of training patterns is 250 training and testing patterns are 120.In the training they obtained 99.79% of accuracy and in Testing they obtained 94.28% of accuracy. These results can predict the rainfall for the future. For rainfall prediction, Artificial Neural Network was applied and the rainfall was predicted in India. According to the results back propagation neural network were acceptably accurate and can be used for predicting the rainfall. So by using this method for prediction we can find the amount of rainfall in the region by using the attributes like humidity, dew point and pressure. Rainfall prediction is one of the most important and challenging task in the modern world. ANN has been successfully used by most of the researchers in this field for the last twenty-five years. [10] Provides a survey of available literature of some methodologies employed by different researchers to utilize ANN for rainfall prediction. From the survey it has been found that most of the researchers used back propagation network for rainfall prediction and got significant results. The survey reports that rainfall prediction using ANN technique is more suitable and also gives a conclusion that the forecasting techniques that use MLP, BPN, RBFN, SOM and SVM are suitable to predict rainfall than other forecasting techniques such as statistical and numerical methods. Kadu, Wagh and Chatur [11] proposed a model of temperature prediction which uses new wireless technology for data gathering with the combination of statistica software. They proposed approach of artificial neural network that uses analysis of data and learn from it for future predictions of temperature, with the combination of wireless technology and statistica software. Another researcher Naik, Pathan [12] proposed a new technique of Weather classification and forecasting using Levenberg Marquardt Back Propagation Feed Forward Neural Network. As there are many BP algorithm but among them Levenberg BP is the fastest. The classification and Prediction of Weather using BPNN is basically a forecasting kit which aims to gather data i.e. weather parameters like temperature, pressure, humidity, wind direction. These predictors are taken as input neuron to BP .Past and Present data of atmosphere is collected and used to train Neural Network. Devi, Reddy, Kumar, etc [13] presented a neural network-based algorithm for predicting the temperature. The Neural Networks package supports different types of training or learning algorithms. One such algorithm is Back Propagation Neural Network (BPN) technique. They tested the proposed idea using the real time dataset. The results were compared with practical working of meteorological department and these results confirm that the model have the potential for successful application to temperature forecasting. [14] Uses Artificial Neural Network (ANN) to predict and classify thunderstorms. ANN has designed to forecasts the occurrence of thunderstorm in two geographical regions. Thus, it is concluded from the results that ANN can be effectively utilized for the prediction and classification of thunderstorm with appreciable level of accuracy. Rankovic, Divac, Nikola etc [15] showed that how ANN can be beneficial to control floods by ensuring safety of Dams. The safety control of dams is based on seepage flows, seepage water clarity, piezometric levels, water levels, pressures, deformations or movements, temperature variations, loading conditions, etc. Interpretation of these large sets of data is very important for dam health monitoring and it is based on mathematical models. They study to develop a feed forward neural network (FNN) model to predict the piezometric water level in dams. An improved resilient propagation algorithm has been used to train the FNN. The measured data have been compared with the results of FNN models and multiple linear regression (MLR) models. The results of this study show that FNN models can be a powerful and important tool which can be used to assess dams. The nonlinear FNN model approach has been shown to provide a better prediction of water levels in piezometers than MLR.

3 Requirements Specification

3.1 Hardware Requirements

Computer for Computational Programming and Analysis.

3.2 Software Requirements

A coding environment for Python and R.

4 System Design

Risk calculation and Prediction of Rain Tomorrow based on classes in input, Classifying based on previous data when it rained or not. Which ones are closer have more chances.

Naïve Bayes or Bayes Classification gave the highest accuracy of 91.81% for rain prediction. Since, different models fit differently for different datasets, Bayes Classification fits best for the above dataset.

5 Implementation of System

For justified comparison parallel models are trained against identical training, test and validation which include:

1. Decision Tree
2. Bayes Classification

3. KNN Classification

Decision Tree

Decision Tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

Decision tree learning employs a divide and conquer strategy by conducting a greedy search to identify the optimal split points within a tree. This process of splitting is then repeated in a top-down, recursive manner until all, or the majority of records have been classified under specific class labels. Whether or not all data points are classified as homogenous sets is largely dependent on the complexity of the decision tree. Smaller trees are more easily able to attain pure leaf nodes—i.e. data points in a single class. However, as a tree grows in size, it becomes increasingly difficult to maintain this purity, and it usually results in too little data falling within a given subtree. When this occurs, it is known as data fragmentation, and it can often lead to overfitting. As a result, decision trees have preference for small trees, which is consistent with the principle of parsimony in Occam's Razor; that is, "entities should not be multiplied beyond necessity." Said differently, decision trees should add complexity only if necessary, as the simplest explanation is often the best. To reduce complexity and prevent overfitting, pruning is usually employed; this is a process, which removes branches that split on features with low importance. The model's fit can then be evaluated through the process of cross-validation. Another way that decision trees can maintain their accuracy is by forming an ensemble via a random forest algorithm; this classifier predicts more accurate results, particularly when the individual trees are uncorrelated with each other.

Bayes Classification

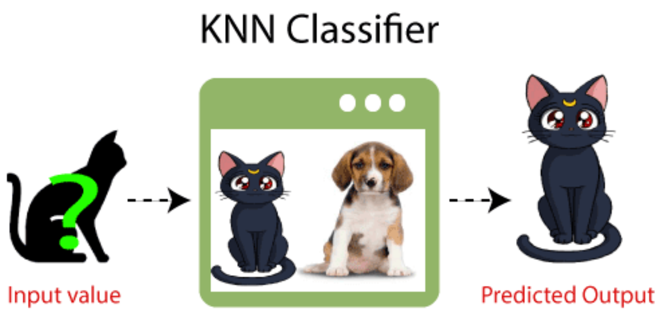
Naive Bayes classifiers are a collection of classification algorithms based on **Bayes' Theorem**. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

KNN Classification

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it

does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN

algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.



6 Results and Discussion

I. Models

i. Decision Tree

Observation: Risk calculation and Prediction of Rain Tomorrow based on classes in input.
Accuracy: 0.7575

ii. Bayes Classification

Observation: Predicting rain based on probability of Raining Today vs Not Raining
Accuracy: 0.9181

iii. KNN Classification

Observation: Classifying based on previous data when it rained or not. Which ones are closer have more chances.
Accuracy:
Three Neighbors: 0.8727
Five Neighbors: 0.9
Seven: 0.8909
Nine: 0.8818
Eleven: 0.8818
Thirteen: 0.8636
Best accuracy is obtained on 5 neighbors
Best Accuracy: 0.9

iv. Regression Tree

Observation: Predicting Rain based on numerical values like Temperature, Pressure, Humidity, etc
Accuracy: 0.8484

Best Model: Supervised
Bayes Classification
Naïve Bayes or Bayes Classification gave the highest accuracy of 91.81% for rain prediction. Since, different models fit differently for different datasets, Bayes Classification fits best for the above dataset.
Accuracy: 0.8484

v. **Hierarchical Clustering**
Agglomerative Clustering
Observation: Since this is unsupervised algorithm, it classifies data into two clusters. Cluster1 (purple) represents the days on which it will not rain, while red dots represent days on which it will rain.

vi. **K-Means**
Observation: Since this is unsupervised algorithm, it classifies data into two clusters. Cluster1 (Blue) represents the days on which it will not rain, while green dots represent days on which it will rain.
This is slightly inaccurate than Agglomerative Clustering

vii. **K- Medoid**
Observation: Since this is unsupervised algorithm, it classifies data into two clusters. Cluster1 (Cyan) represents the days on which it will not rain, while violet dots represent days on which it will rain.
Here, we can see that some points are mixed together. Those points are exceptions or outliers.

II. Plots

i. **Temperature vs Evaporation**
Observation: Evaporation increases with Temperature.

ii. **Temperature Range vs Wind Speed**
Observation: As temperature difference increases wind speed increases.

iii. **Pressure vs Rainfall**
Observation: As pressure increases, rainfall decreases.

iv. **Rainfall and Frequency**
Observation: Frequency of rainfall in dataset.

v. **Risk and Frequency**
Observation: Shows relation between Risk calculated and observed**.**

vi. **Logistic Regression**
Observation: Predicts probability of rain from given attributes. Here, we gave Evaporation and Wind speed as attributes.

7. Conclusion and Future Work

From the models we implemented, the best accuracy was given by bayes classification (91%).
So our algorithm is accurate and can be implemented to predict Chennai city weather.

8. REFERENCES

- https://www.kaggle.com/datasets/mahendran1/weather-data-in-india-from-1901-to-2017
- https://cdsp.imdpune.gov.in/
- https://datasetsearch.research.google.com/

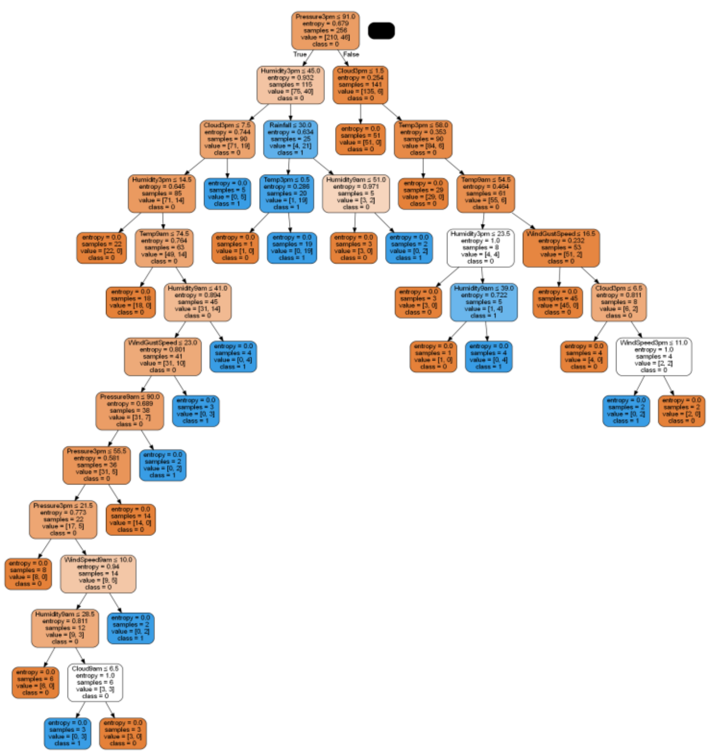
APPENDIX

I. Models

Model1: Decision Tree

Code:

```
from sklearn.tree import DecisionTreeClassifier
rain = DecisionTreeClassifier(criterion="entropy", random_state=100)
rain.fit(x_train, y_train)
```



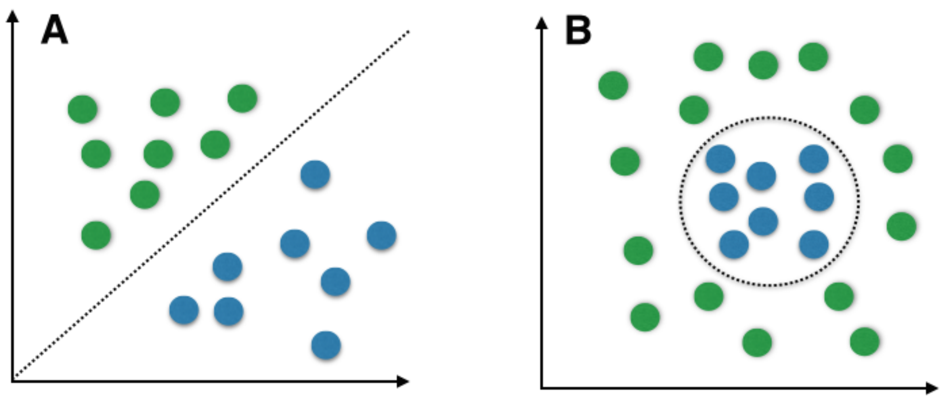
Model2: Bayes Classification

Code:

From sklearn.naive_bayes import GaussianNB

rain = GaussionNB()

rain.fit()(x_train, y_train)



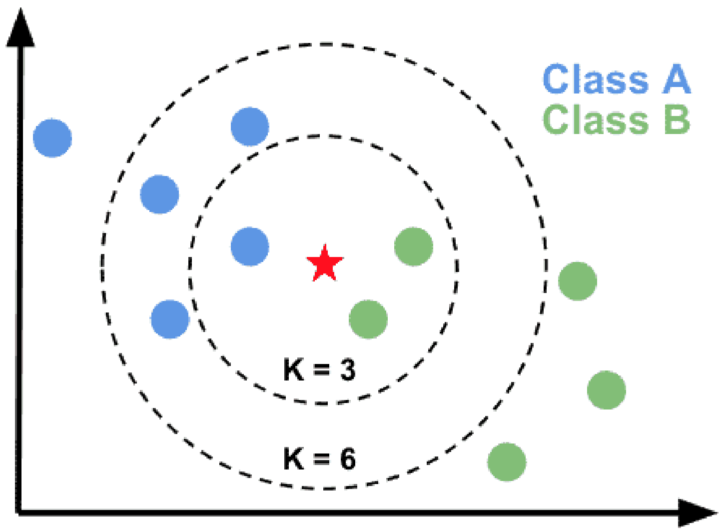
Model3:KNN Classification

Code:

from sklearn.neighbors import KNeighborsClassifier

classifier = KNeighborsClassifier(n_neighbors=5)

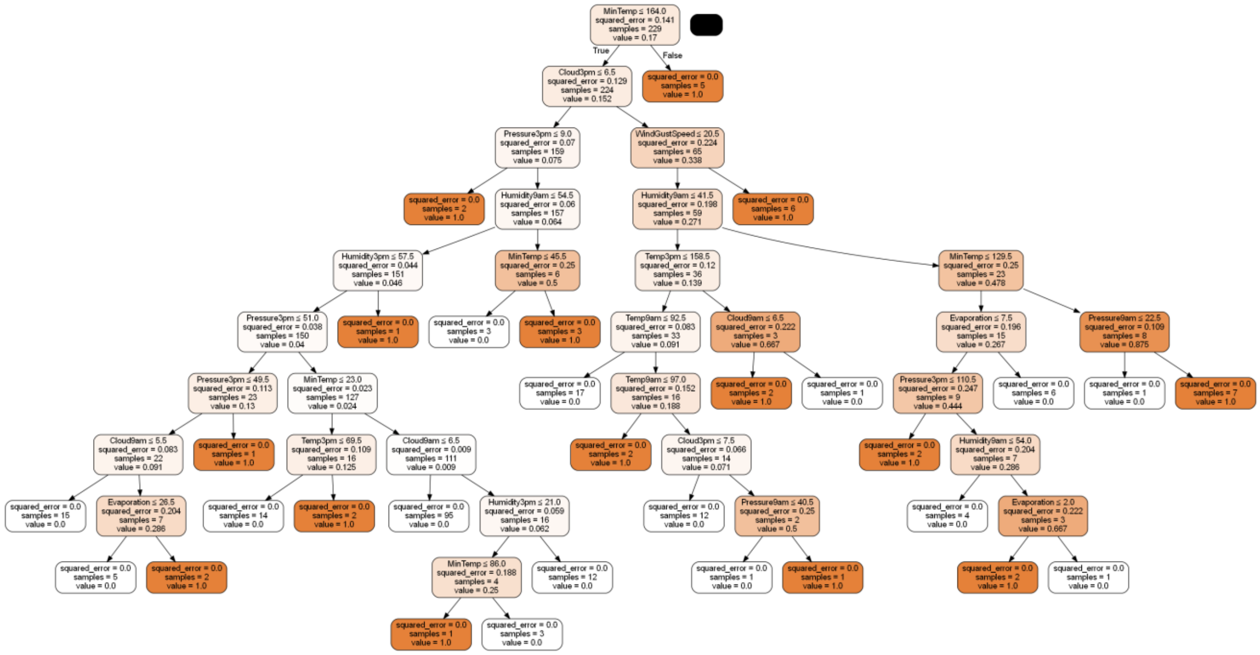
classifier.fit(x_train, y_train)



Model4:Regression Tree

Code:

```
from sklearn.tree import DecisionTreeRegressor
rain = DecisionTreeRegressor(random_state=100)
rain.fit(x_train, y_train)
```

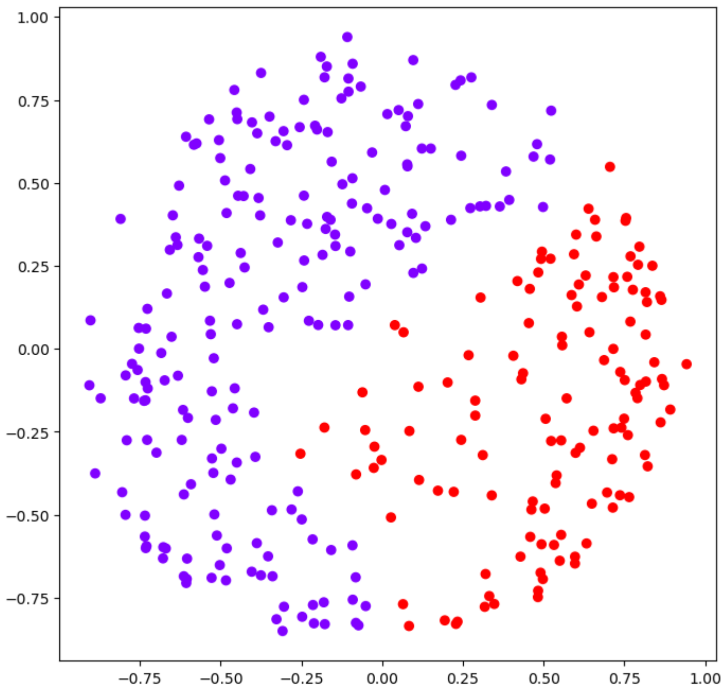


Model5: Heirarchical Clustering

Code:

```
agg2 = AgglomerativeClustering(n_clusters = 2)
plt.figure(figsize = (8,8))
plt.scatter(df_prp['X'], df_prp['Y'], c = agg2.fit_predict(df_prp), cmap = 'rainbow')
```

plt.show()



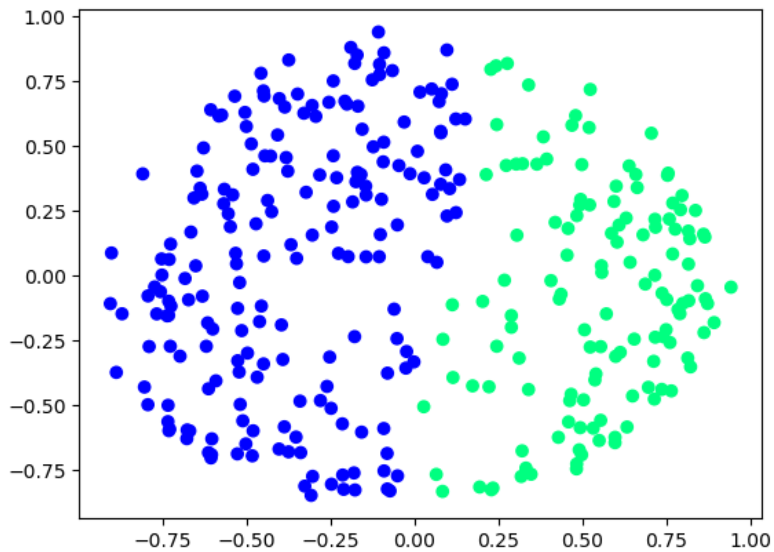
Model6: K-Means

Code:

from sklearn.cluster import KMeans

kmeans2 = KMeans(n_clusters=2, random_state=100)

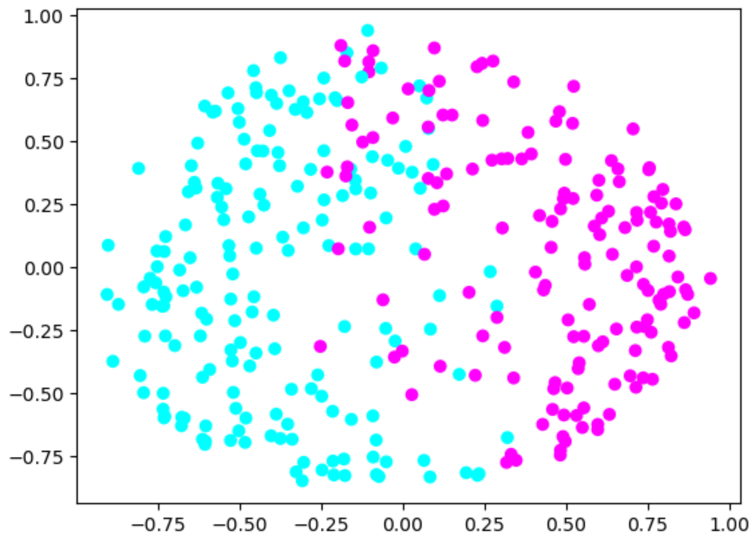
kmeans2.fit(df)



Model7: K-Mediod

Code:

```
from sklearn_extra.cluster import KMedoids
kmedo2 = KMedoids(n_clusters = 2, random_state=1)
kmedo2.fit(df)
```



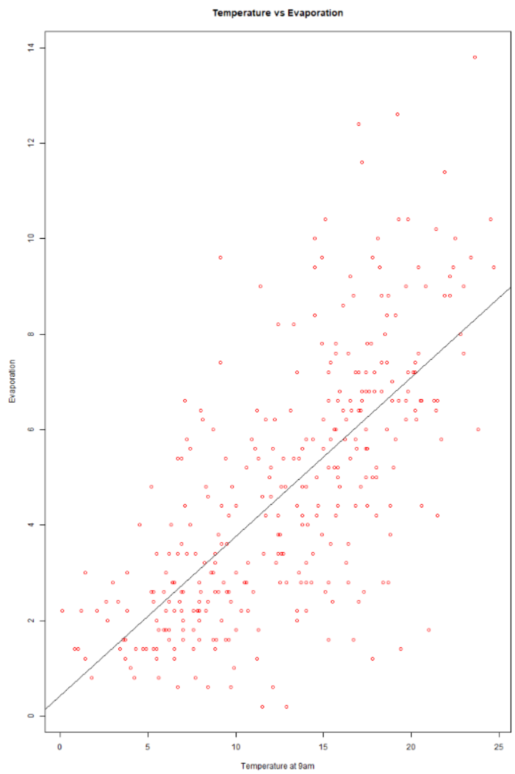
II. Plots

Temperature vs Evaporation

Code:

plot(x=df$Temp9am, y=df$Evaporation,col="red",xlab="Temperature at 9am",ylab="Evaporation", main="Temperature vs Evaporation")

abline(lm(df$Evaporation ~ df$Temp9am,col="green"))



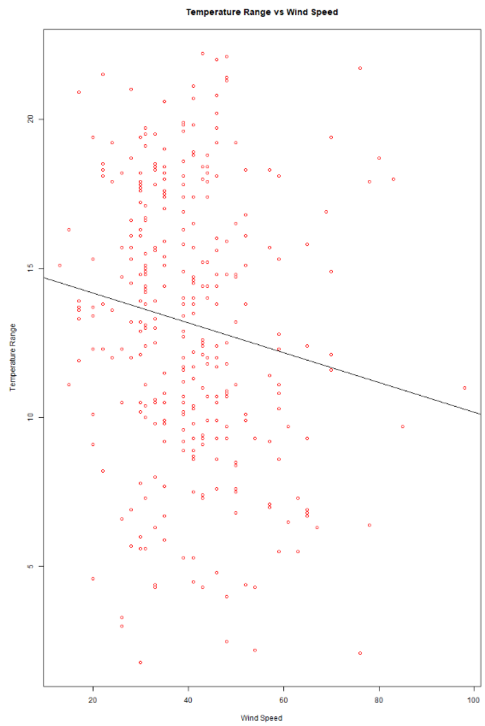
Temperature vs Wind Speed

Code:

df$Range = df$MaxTemp - df$MinTemp;

plot(y=df$Range, x=df$WindGustSpeed, col="red",xlab="Wind Speed",ylab="Temperature Range", main="Temperature Range vs Wind Speed")

abline(lm(df$Range~ df$WindGustSpeed ,col="green"))

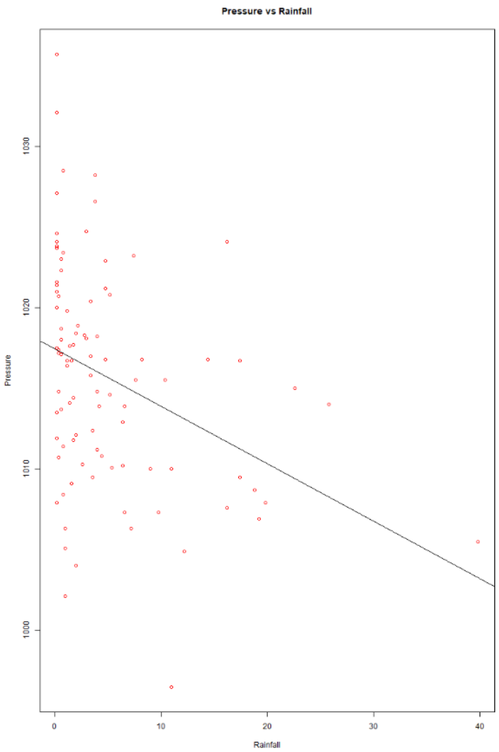


Pressure vs Rainfall

Code:

plot(y=p, x=v, col="red", xlab="Rainfall", ylab="Pressure", main="Pressure vs Rainfall")

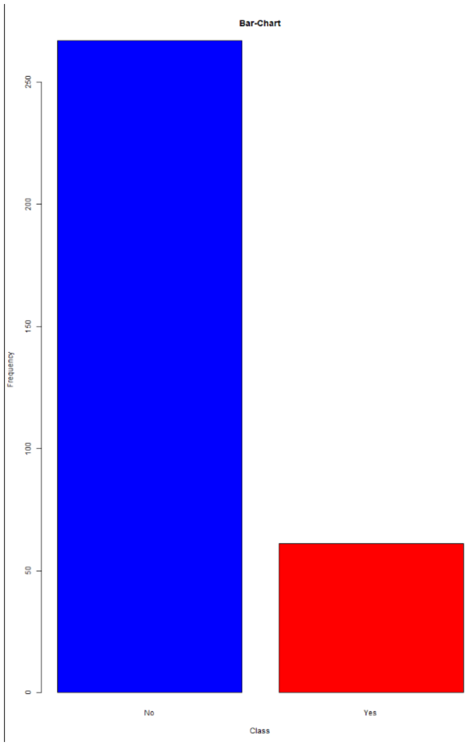
abline(lm(p ~ v))



Rainfall and Frequency

Code:

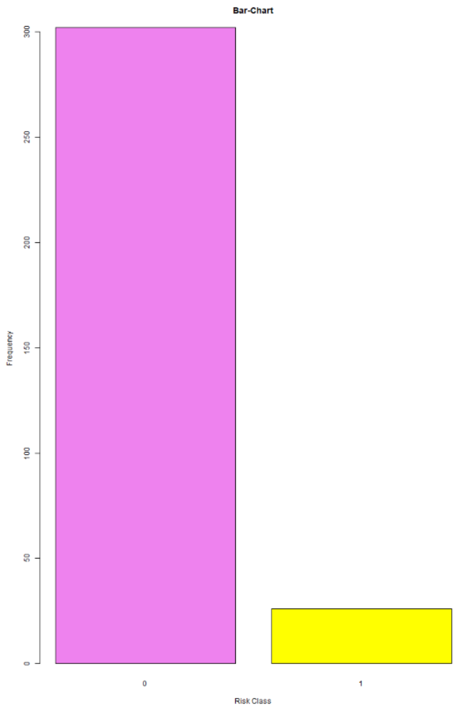
barplot(table(df$RainToday), xlab = "Class", ylab = "Frequency", main ="Bar-Chart",col=c("blue","red"))



Risk and Frequency

Code:

barplot(table(df$RainTomorrow), xlab = "X-axis", ylab = "Y-axis", main ="Bar-Chart",col=c("brown","green"))



Logistic Regression

Code:

```
plot(y=df$Risk_Class, x=df$Evaporation+df$Rainfall, col="red",xlab="Rainfall+Evap",ylab="Rain Tomorrow chances", main="Rain Tomorrow probability")
abline(glm(df$Risk_Class ~ df$Evaporation + df$Rainfall))
```

abline(h=0.5, col=("blue"))