A Mini Project Report

on

Weather Prediction Using Markov Chain

In Subject: **Probability & Statistics**

by

Yash Akotkar (22010814)
Rajdeep Chaurasia (22010948)
Netal Daga (22010244)
Samarth Ghule (22010468)

Department of Artificial Intelligence and Data Science

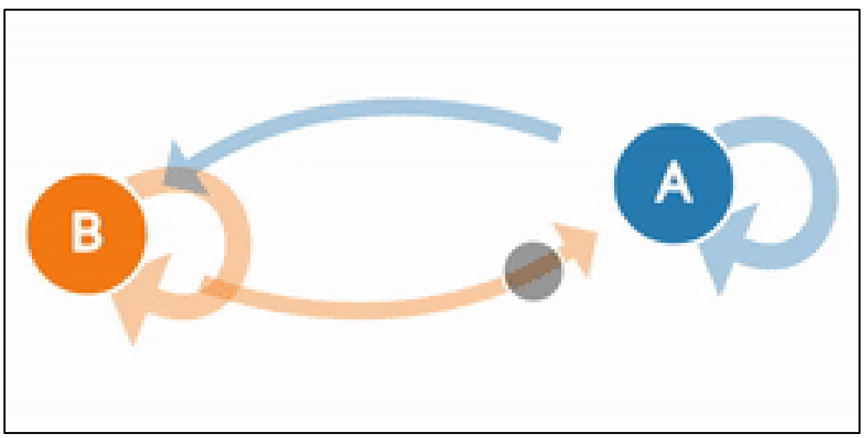
VIIT

2021-2022

Contents

1.1 Introduction

- In this PBL we have tried to build a project using concepts of Markov chain for weather prediction.
- The complete project is performed using R language on R - studio.
- This is very primitive idea which will help us to understand the theoretical concepts and implement them practically.
- **Stochastic process** is the process of some values changing randomly over time. At its simplest form, it involves a variable changing at a random rate through time. There are various types of stochastic processes. Some well-known types are random walks, Markov chains, and Bernoulli processes. They are used in mathematics, engineering, computer science, and various other fields. They can be classified into two distinct types: discrete-time and continuous stochastic processes.
- The %*% operator is a special kind of multiplication operator. It is used in the multiplication of matrices in R.
- A Markov chain is a mathematical system that experiences transitions from one state to another according to certain probabilistic rules. The defining characteristic of a Markov chain is that no matter how the process arrived at its present state, the possible future states are fixed.
- Overall, Markov Chains are conceptually quite intuitive, and are very accessible in that they can be implemented without the use of any advanced statistical or mathematical concepts. They are a great way to start learning about probabilistic modelling and data science techniques.



- A Markov chain is a stochastic process, but it differs from a general stochastic process in that a Markov chain must be "memory-less. " That is, (the probability of) future actions are not dependent upon the steps that led up to the present state. This is called the Markov property.

Types of Markov Chain :

- ***discrete-time Markov chains*** : This implies the index set T(state of the process at time t) is a countable set here or we can say that changes occur at specific states. Generally, the term “Markov chain” is used for DTMC.
- ***continuous-time Markov chains***: Here the index set T(state of the process at time t) is a continuum, which means changes are continuous in CTMC.

Properties of Markov Chain :

- A Markov chain is said to be ***Irreducible*** if we can go from one state to another in a single or more than one step.
- A state in a Markov chain is said to be ***Periodic*** if returning to it requires a multiple of some integer larger than 1, the greatest common divisor of all the possible return path lengths will be the period of that state.
- A state in a Markov chain is said to be ***Transient*** if there is a non-zero probability that the chain will never return to the same state, otherwise, it is Recurrent.
- A state in a Markov chain is called ***Absorbing*** if there is no possible way to leave that state. Absorbing states do not have any outgoing transitions from it.

1.2 Requirements

- Basic knowledge of programming & Markov chain.
- Software like R-studio.

1.3 Problem Statement

Weather prediction using Markov chain, transition matrix, Markov property in R-studio.

1.4 Proposed Work

```
library(markovchain)

library(diagram)

library(expm)

tmA <- matrix (c (0.25,0.65,0.1,.25,0.25,.5,.35,.25,0.4), nrow = 3, byrow = TRUE)

dtmcA <- new ("markovchain", transitionMatrix=tmA, states=c ("No Rain", "Light
Rain”, “Heavy Rain"), name="MarkovChain A")

dtmcA

plot(dtmcA)

stateNames <- c ("No Rain", "Light Rain", "Heavy Rain")

row.names (tmA) <- stateNames; colnames(tmA) <- stateNames

plotmat (tmA,pos = c(1,2),

        lwd = 1, box.lwd = 2,

        cex.txt = 0.8,

        box.size = 0.1,

        box.type = "circle",

        box.prop = 0.5,

        box.col = "light blue",

        arr.length=.1,

        arr.width=.1,

        self.cex = .6,

        self.shifty = -.01,

        self.shiftx = .14,

        main = "Markov Chain")

initialState<-c(0,1,0)
```

```
steps<-2
finalState<-initialState*dtmcA^steps #using power operator
finalState
steadyStates(dtmcA)

data(rain)
rain
mysequence<-rain$rain
createSequenceMatrix(mysequence)
head(rain)
myFit<-markovchainFit(data=mysequence,confidencelevel = .9,method = "mle")
myFit
alofiMc<-myFit$estimate
alofiMc
a11=alofiMc[1,1]
a12=alofiMc[1,2]
a13=alofiMc[1,3]
a21=alofiMc[2,1]
a22=alofiMc[2,2]
a23=alofiMc[2,3]
a31=alofiMc[3,1]
a32=alofiMc[3,2]
a33=alofiMc[3,3]

## Hard code the transition matrix
```

```
stateNames <- c("No Rain","Light Rain","Heavy Rain")

ra <- matrix(c(a11,a12,a13,a21,a22,a23,a31,a32,a33),nrow=3, byrow=TRUE)

#ra <- matrix(c(0.660,0.230,0.110,0.463,0.306,0.231,0.198,0.312,0.490),nrow=3,
byrow=TRUE)


dtmcA <- new("markovchain",transitionMatrix=ra, states=c("No Rain","Light
Rain","Heavy Rain"), name="MarkovChain A")


dtmcA

plot(dtmcA)

row.names(ra) <- stateNames; colnames(ra) <- stateNames

ra = round(ra,3)

plotmat(ra,pos = c(1,2),

        lwd = 1, box.lwd = 2,

        cex.txt = 0.8,

        box.size = 0.1,

        box.type = "circle",

        box.prop = 0.5,

        box.col = "light blue",

        arr.length=.1,

        arr.width=.1,

        self.cex = .4,

        self.shifty = -.01,

        self.shiftx = .13,

        main = "Markov Chain Transition Matrix")
```

```
x1 <- matrix(c(1,0,0),nrow=1, byrow=TRUE)
x1 %*% ra
ra2 <- ra %^% 2
ra3 <- ra %^% 3
ra4 <- ra %^% 4
ra5 <- ra %^% 5
ra6 <- ra %^% 6
ra7 <- ra %^% 7
cat("Day 1 Forecast")
round(x1%*%ra,3)
cat("Day 2 Forecast")
round(x1%*%ra2,3)
cat("Day 3 Forecast")
round(x1%*%ra3,3)
cat("Day 4 Forecast")
round(x1%*%ra4,3)
cat("Day 5 Forecast")
round(x1%*%ra5,3)
cat("Day 6 Forecast")
round(x1%*%ra6,3)
cat("Day 7 Forecast")
round(x1%*%ra7,3)
ra7=round(ra7,3)
plotmat(ra7,pos = c(1,2),
    lwd = 1, box.lwd = 2,
```

```
cex.txt = 0.8,
box.size = 0.1,
box.type = "circle",
box.prop = 0.5,
box.col = "light blue",
arr.length=.1,
arr.width=.1,
self.cex = .6,
self.shifty = -.01,
self.shiftx = .14,
main = "The transition matrix after 7 days")
```

2.1 Approach

First learnt R-studio & related libraries:

- library(markovchain): R package providing classes, methods and function for easily handling Discrete Time Markov Chains (DTMC), performing probabilistic analysis and fitting.
- library(diagram): For plotting the diagrams.
- library(expm): For exponential.
- createSequenceMatrix: returns a function showing previous vs actual states from the pairs in a given sequence.
- markovchainFit: function allows to obtain the estimated transition matric and the confidence levels (using elliptic MLE hyphotesis).
- Matlab: that contains functions for matrix management and calculations that emulate those within Matlab environment

Here is a fictional example cited by Berchtold, (1998) of a random variable Xt represents the meteorological situation of a city over a period of 20 days (i.e) t = 1,

..., 20. This variable takes three modalities: E = {Rainy, Covered, Sunny} the 20 successive observations of xt: R S C C C R C C S S S C R C C R C C C S.

$$\begin{array}{cc} & \begin{array}{ccc} R & C & S \end{array} \\ T = \begin{array}{c} R \\ C \\ S \end{array} & \begin{pmatrix} 0 & 3 & 1 \\ 3 & 6 & 2 \\ 0 & 2 & 2 \end{pmatrix} \end{array} \quad \begin{array}{c} Total \\ \begin{pmatrix} 4 \\ 11 \\ 4 \end{pmatrix} \end{array}$$

We build a contingency table that describes the relationship between two successive observations of the Xt variable.

The transformation of these tables into matrices of transition then in Markov chain will allow us to add dynamic and predictive dimensions.

by dividing each element by the total of the corresponding line, we find a matrix called : transition matrix :

$$\begin{array}{cc} & \begin{array}{ccc} R & C & S \end{array} \\ \boldsymbol{P} = \begin{array}{c} R \\ C \\ S \end{array} & \begin{pmatrix} 0 & 0.75 & 0.25 \\ 0.27 & 0.55 & 0.18 \\ 0 & 0.5 & 0.5 \end{pmatrix} \end{array} \quad \begin{array}{c} Total \\ \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \end{array}$$

A transition matrix P(t−1;t) of dimension m timesm, is the one that summarizes the probabilities pi,j(t), corresponding to the different possible states of i and j is :

$$P_{(t-1;t)}^{(m,m)} = \begin{pmatrix} p_{1,1}(t) & p_{1,2}(t) & \cdots & p_{1,m}(t) \\ p_{2,1}(t) & p_{2,2}(t) & \cdots & p_{2,m}(t) \\ \vdots & \vdots & \ddots & \vdots \\ p_{m,1}(t) & p_{m,2}(t) & \cdots & p_{m,m}(t) \end{pmatrix}$$

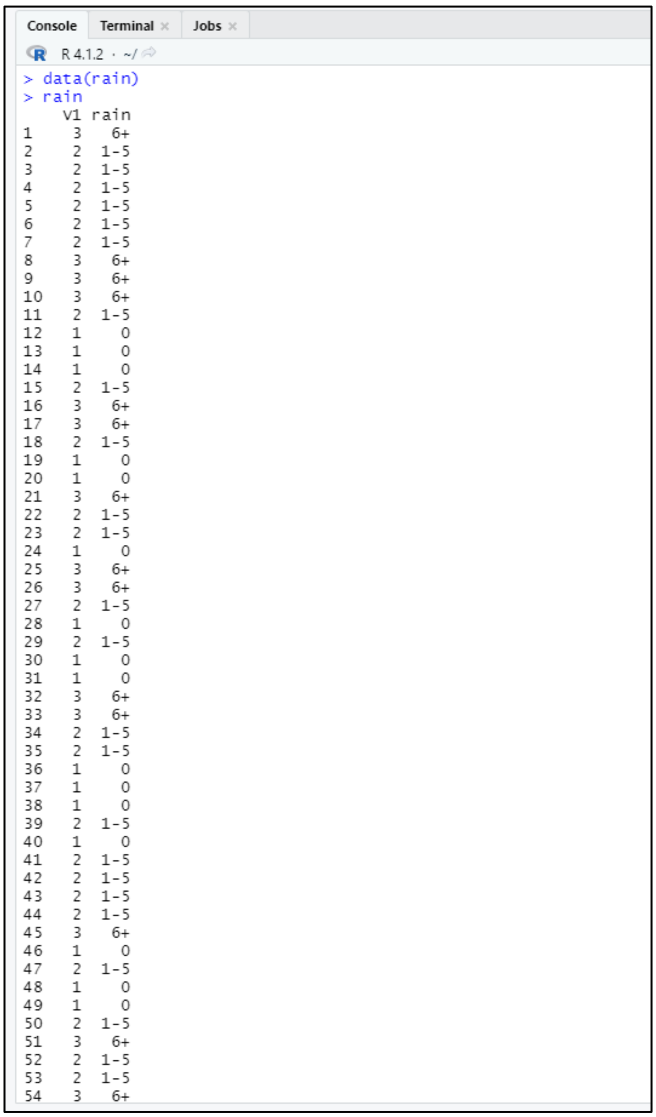
All the lines of this matrix are probability distributions, (i.e.) the sum of the coefficients of each line is equal to 1.

2.2 Platform & Technology

Platform used is [R-studio]() and technology used is [R language.]()

2.3 Outcomes & Working

1.) Reading Dataset:



```
Console  Terminal  Jobs
R 4.1.2 · ~/
> data(rain)
> rain
   V1 rain
1   3   6+
2   2  1-5
3   2  1-5
4   2  1-5
5   2  1-5
6   2  1-5
7   2  1-5
8   3   6+
9   3   6+
10  3   6+
11  2  1-5
12  1    0
13  1    0
14  1    0
15  2  1-5
16  3   6+
17  3   6+
18  2  1-5
19  1    0
20  1    0
21  3   6+
22  2  1-5
23  2  1-5
24  1    0
25  3   6+
26  3   6+
27  2  1-5
28  1    0
29  2  1-5
30  1    0
31  1    0
32  3   6+
33  3   6+
34  2  1-5
35  2  1-5
36  1    0
37  1    0
38  1    0
39  2  1-5
40  1    0
41  2  1-5
42  2  1-5
43  2  1-5
44  2  1-5
45  3   6+
46  1    0
47  2  1-5
48  1    0
49  1    0
50  2  1-5
51  3   6+
52  2  1-5
53  2  1-5
54  3   6+
```

2.) Creating sequence:

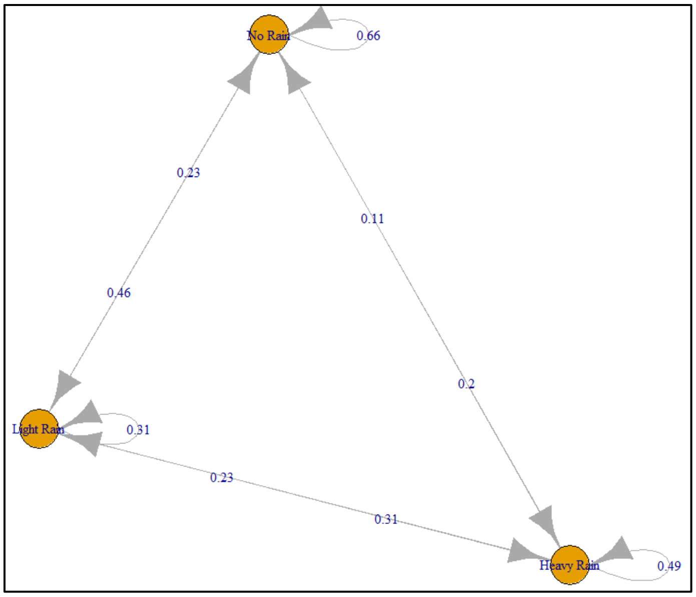
```
> mysequence<-rain$rain
> createSequenceMatrix(mysequence)
      0 1-5  6+
0   362 126  60
1-5 136  90  68
6+   50  79 124
> head(rain)
  V1 rain
1  3   6+
2  2  1-5
3  2  1-5
4  2  1-5
5  2  1-5
6  2  1-5
>
```

3.) Using MLE model:

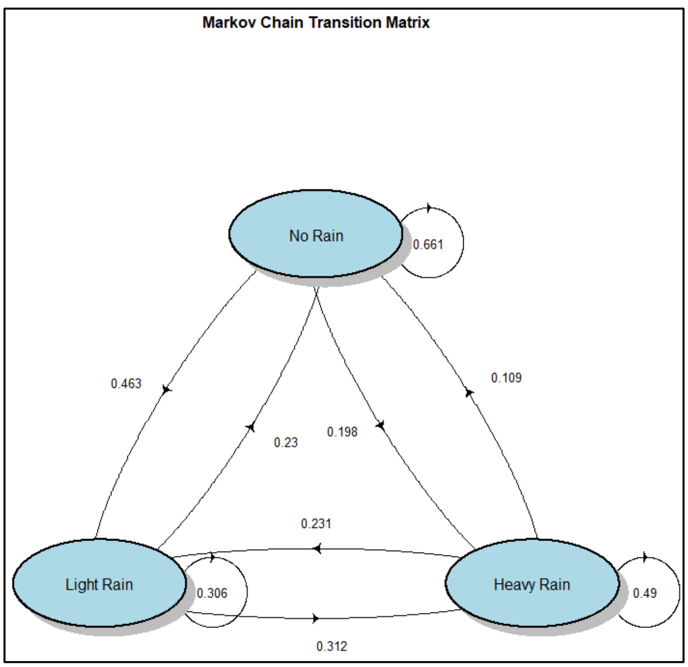
```
> myFit<-markovchainFit(data=mysequence,confidencelevel = .9,method = "mle")
> myFit
$estimate
MLE Fit
 A  3 - dimensional discrete Markov Chain defined by the following states:
 0, 1-5, 6+
 The transition matrix  (by rows)  is defined as follows:
            0       1-5        6+
0   0.6605839 0.2299270 0.1094891
1-5 0.4625850 0.3061224 0.2312925
6+  0.1976285 0.3122530 0.4901186


$standardError
             0        1-5         6+
0   0.03471952 0.02048353 0.01413498
1-5 0.03966634 0.03226814 0.02804834
6+  0.02794888 0.03513120 0.04401395

$confidenceLevel
[1] 0.9

$lowerEndpointMatrix
            0       1-5         6+
0   0.6034754 0.1962346 0.08623909
1-5 0.3973397 0.2530461 0.18515711
6+  0.1516566 0.2544673 0.41772208

$upperEndpointMatrix
            0       1-5        6+
0   0.7176925 0.2636194 0.1327390
1-5 0.5278304 0.3591988 0.2774279
6+  0.2436003 0.3700386 0.5625151

$logLikelihood
[1] -1040.419

>
```

4.) dtmcA Transition matrix:

```
> dtmcA
MarkovChain A
 A  3 - dimensional discrete Markov Chain defined by the following states:
 No Rain, Light Rain, Heavy Rain
 The transition matrix  (by rows)  is defined as follows:
            No Rain Light Rain Heavy Rain
No Rain    0.6605839  0.2299270  0.1094891
Light Rain 0.4625850  0.3061224  0.2312925
Heavy Rain 0.1976285  0.3122530  0.4901186
```



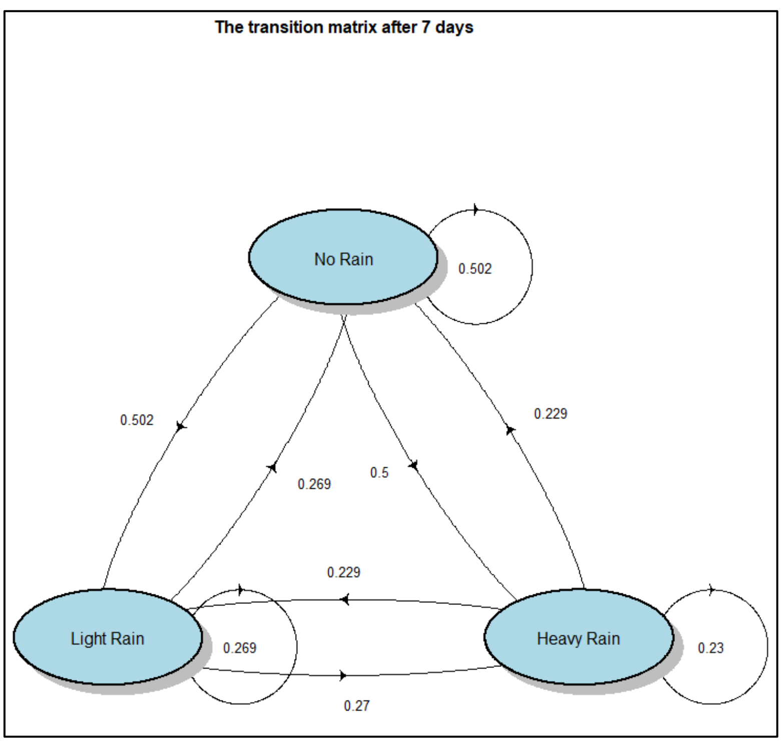
5.) Markov chain transition matrix:



6.) 7 days forecasting:

```
      No Rain Light Rain Heavy Rain
[1,]    0.661       0.23      0.109
> ra2 <- ra %^% 2
> ra3 <- ra %^% 3
> ra4 <- ra %^% 4
> ra5 <- ra %^% 5
> ra6 <- ra %^% 6
> ra7 <- ra %^% 7
> cat("Day 1 Forecast")
Day 1 Forecast> round(x1%*%ra,3)
      No Rain Light Rain Heavy Rain
[1,]    0.661       0.23      0.109
> cat("Day 2 Forecast")
Day 2 Forecast> round(x1%*%ra2,3)
      No Rain Light Rain Heavy Rain
[1,]    0.565      0.256      0.179
> cat("Day 3 Forecast")
Day 3 Forecast> round(x1%*%ra3,3)
      No Rain Light Rain Heavy Rain
[1,]    0.528      0.264      0.208
> cat("Day 4 Forecast")
Day 4 Forecast> round(x1%*%ra4,3)
      No Rain Light Rain Heavy Rain
[1,]    0.512      0.267      0.221
> cat("Day 5 Forecast")
Day 5 Forecast> round(x1%*%ra5,3)
      No Rain Light Rain Heavy Rain
[1,]    0.506      0.268      0.226
> cat("Day 6 Forecast")
Day 6 Forecast> round(x1%*%ra6,3)
      No Rain Light Rain Heavy Rain
[1,]    0.503      0.269      0.228
> cat("Day 7 Forecast")
Day 7 Forecast> round(x1%*%ra7,3)
      No Rain Light Rain Heavy Rain
[1,]    0.502      0.269      0.229
> ra7=round(ra7,3)
>
```

7.) Transition matrix after 7 days:



• PBL Group 2 •

PAGE NO.: 01
DATE 3/05/22

Working on discussed example...

- Here is a meteorogical situation of a city over period of 20 days. Define any current state & predict tomorrow's state. Three modalities are : $E = \{Rainy(R), Covered(C), Sunny(S)\}$

Data :

R S C C C R C C C S S S C R C C R C C C S

Solⁿ: Firstly we need to create a sequence matrix;

$$P = \begin{array}{c|ccc} & R & C & S \\ R & 0 & 3 & 1 \\ C & 3 & 6 & 2 \\ S & 0 & 2 & 2 \end{array}$$

⇒ this matrix is generated by counting the sequence in given data.

$$T = \begin{bmatrix} 4 \\ 11 \\ 4 \end{bmatrix}$$ ⇒ sum of elements (row wise)

To create transition/stochastic matrix we need to divide sequence matrix with total matrix (row wise)

i.e. $0/4 = 0$ $\quad 3/11 = 0.27$ $\quad 0/4 = 0$

$3/4 = 0.75$ $\quad 6/11 = 0.55$ $\quad 2/4 = 0.5$

$1/4 = 0.25$ $\quad 2/11 = 0.18$ $\quad 2/4 = 0.5$

$$P = \begin{array}{c|ccc} & R & C & S \\ R & 0 & 0.75 & 0.25 \\ C & 0.27 & 0.55 & 0.18 \\ S & 0 & 0.5 & 0.5 \end{array}$$

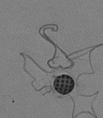
We have defined current stated as 'covered' so the initial state matrix will be $(0, 1, 0)$

For predicting tomorrow's state i.e. final state we use formula,

$$\text{final state} = \text{initial state} \times (SM)^{steps}$$

Here, initial state $= [0\ 1\ 0]$

$$SM = \begin{bmatrix} 0 & 0.75 & 0.25 \\ 0.27 & 0.55 & 0.18 \\ 0.00 & 0.50 & 0.50 \end{bmatrix}$$



step = 1 (as we are predicting next day)

$$\therefore \text{final state} = \underbrace{[0\ 1\ 0] \begin{bmatrix} 0 & 0.75 & 0.25 \\ 0.27 & 0.55 & 0.18 \\ 0 & 0.50 & 0.50 \end{bmatrix}}_{\text{Matrix Multiplication}} = \begin{array}{ccc} R & C & S \\ \end{array} [0.27\ 0.55\ 0.18]$$

Hence, tomorrow it can be 27% rainy, 55% coverred and 18% sunny.

2.4 Future Work

We can do weather prediction by considering current heat degree from different states & predict it.

2.5 Challenges

1.) Choosing the right dataset.

2.) Understanding the packages and various functions.

3.) Determining the train and test data at various stages of the project.

4.) Understanding the minute details of the different graphs and plots.

Conclusion:

Now that you know the basics of Markov chains, we should now be able to easily implement them in a language of our choice. In our opinion, the natural progression along the theory route would be toward Hidden Markov Processes. Simple Markov chains are the building blocks of other, more sophisticated, modeling techniques, so with this knowledge, you can now move onto various techniques within topics such as belief modeling and sampling.

References:

Probability & statistics Notes.

Reference Research Paper:
https://www.researchgate.net/publication/336498129_Markov_Chain_Analysis_With_R_A_Brief_Introduction

THANK YOU!!