Seasonal

Ricky Trujillo

2024-05-09

```
weather_summary = read.csv("weather.csv")
weather_summary$weather_state = as.factor(weather_summary$weather_state)

states <- c("Chilly", "Cloudy", "Partly Cloudy", "Rainy", "Sunny")
weather_summary$prediction<- rep(NA,nrow(weather_summary))
weather_summary$prediction = factor(weather_summary$prediction, levels= states)


weather24_pred = weather_summary %>%
  filter(Date>= as.Date("2024-01-01", "%Y-%m-%d"))%>%
  select(Date, Month, weather_state, prediction)

state_mat = one_hot(as.data.table(weather_summary$weather_state))
colnames(state_mat)<-  states
```

```
weather_seasons = weather_summary%>%
  mutate(Season = ifelse(Month %in% c(12,1,2), "Winter",
                         ifelse(Month %in% c(3,4,5), "Spring",
                                ifelse(Month %in% c(6,7,8), "Summer",
                                       ifelse(Month %in% c(9,10,11), "Fall", "")))))

Winter = weather_seasons %>%
  filter(Season == "Winter") %>%
  select(-Season)
Winter_1 = Winter[Winter$Month==12, ]
Winter_2 = Winter[Winter$Month!=12, ]
Winter = rbind(Winter_1, Winter_2)

Spring = weather_seasons %>%
  filter(Season == "Spring") %>%
  select(-Season)

Summer = weather_seasons %>%
  filter(Season == "Summer") %>%
  select(-Season)

Fall = weather_seasons %>%
  filter(Season == "Fall") %>%
  select(-Season)
```

```
winter_tm = createSequenceMatrix(Winter$weather_state, toRowProbs = TRUE, possibleStates = states)
winter_tm = as.matrix(winter_tm, nrow=5, ncol=5)
t(round(winter_tm,2))
```

```
##               Chilly Cloudy Partly Cloudy Rainy Sunny
## Chilly          0.72   0.32           0.0  0.39  0.35
## Cloudy          0.18   0.48           0.4  0.22  0.24
## Partly Cloudy   0.02   0.02           0.4  0.00  0.00
## Rainy           0.03   0.15           0.0  0.39  0.00
## Sunny           0.05   0.03           0.2  0.00  0.41
```

```
spring_tm = createSequenceMatrix(Spring$weather_state, toRowProbs = TRUE, possibleStates = states)
spring_tm = as.matrix(spring_tm, nrow=5, ncol=5)
t(round(spring_tm,2))
```

```
##               Chilly Cloudy Partly Cloudy Rainy Sunny
## Chilly          0.69   0.33             0  0.64  0.30
## Cloudy          0.19   0.56             0  0.36  0.12
## Partly Cloudy   0.00   0.00             0  0.00  0.00
## Rainy           0.04   0.07             0  0.00  0.03
## Sunny           0.09   0.04             0  0.00  0.55
```

```
summer_tm = createSequenceMatrix(Summer$weather_state, toRowProbs = TRUE, possibleStates = states)
summer_tm = as.matrix(summer_tm, nrow=5, ncol=5)
t(round(summer_tm,2))
```

```
##               Chilly Cloudy Partly Cloudy Rainy Sunny
## Chilly          0.33   0.09          0.00     0  0.01
## Cloudy          0.17   0.37          0.31     0  0.10
## Partly Cloudy   0.00   0.29          0.19     1  0.14
## Rainy           0.00   0.03          0.00     0  0.00
## Sunny           0.50   0.23          0.50     0  0.75
```

```
fall_tm = createSequenceMatrix(Fall$weather_state, toRowProbs = TRUE, possibleStates = states)
fall_tm = as.matrix(fall_tm, nrow=5, ncol=5)
t(round(fall_tm,2))
```

```
##               Chilly Cloudy Partly Cloudy Rainy Sunny
## Chilly          0.59   0.15          0.08   0.5  0.05
## Cloudy          0.17   0.48          0.38   0.0  0.08
## Partly Cloudy   0.00   0.06          0.15   0.5  0.08
## Rainy           0.00   0.03          0.00   0.0  0.01
## Sunny           0.24   0.27          0.38   0.0  0.79
```

```
initial_state <- c(1,0,0,0,0)
days = 1000
cycles = floor(days/365)
remainder = mod(days, 365)

# Winter: 90
# Spring: 92
# Summer: 92
# Fall:  91

# Cycles=0 meaning it is under 1 year
if(cycles ==0){
  if(remainder<=90){
    final_state = initial_state %*% winter_tm %^%(remainder)
  }else if(remainder>90 & remainder<=182){
    final_state = initial_state %*% (1/2)*(t(winter_tm) %^%(90) +
                                      spring_tm%^%(remainder-90))
```

```
  }else if(remainder>182 & remainder<=274){
    final_state = initial_state %*% (1/3)*(winter_tm %^%(90) +
                                          spring_tm%^%(92)+
                                          summer_tm %^% (remainder-182))
  }else if(remainder>274 & remainder<=364){
    final_state = initial_state %*% (1/4)*(winter_tm %^%(90) +
                                          spring_tm%^%(92)+
                                          summer_tm %^% (92)+
                                          fall_tm %^% (remainder-274))
  }
# cycles >=1 more than 365 days
}else if(cycles >=1){
  if(remainder<=90){
    s1=1
    s2=s3=s4=0
    final_state = initial_state %*% ((1/4)*winter_tm %^%(90*cycles + s1*remainder)+
                                       (1/4)*spring_tm %^% (92*cycles+ s2*(remainder-90))+
                                       (1/4)*summer_tm %^% (92*cycles + s3*(remainder-182))+
                                       (1/4)*fall_tm %^% (91*cycles + s4*(remainder-274)))
  }else if(remainder>90 & remainder<=182){
    s2=1
    s1=s3=s4=0
    final_state = initial_state %*% ((1/4)*winter_tm %^%(90(cycles+1) + s1*remainder)+
                                       (1/4)*spring_tm %^% (92*cycles + s2*(remainder-90))+
                                       (1/4)*summer_tm %^% (92*cycles + s3*(remainder-182))+
                                       (1/4)*fall_tm %^% (91*cycles + s4*(remainder-274)))
  }else if(remainder>182 & remainder<=274){
    s3=1
    s1=s2=s4=0
    final_state = initial_state %*% ((1/4)*winter_tm %^%(90*(cycles+1) + s1*remainder)+
                                       (1/4)*spring_tm %^% (92*(cycles+1) + s2*(remainder-90))+
                                       (1/4)*summer_tm %^% (92*cycles + s3*(remainder-182))+
                                       (1/4)*fall_tm %^% (91*cycles + s4*(remainder-274)))
  }else if(remainder>274 & remainder<=364){
    s4=1
    s1=s2=s3=0
    final_state = initial_state %*% ((1/4)*winter_tm %^%(90*(cycles+1) + s1*remainder)+
                                       (1/4)*spring_tm %^% (92*(cycles+1)+ s2*(remainder-90))+
                                       (1/4)*summer_tm %^% (92*(cycles+1) + s3*(remainder-182))+
                                       (1/4)*fall_tm %^% (91*cycles + s4*(remainder-274)))
  }

}

final_state
```

```
##        Chilly    Cloudy Partly Cloudy      Rainy     Sunny
## [1,] 0.316596 0.2377469    0.06683213 0.03952388 0.3393011
```

```
seasonal <- function(weather24_pred){
  prob = list()

  # First entry
  prob[[1]] = t(winter_tm) %*% t(state_mat[730,])
```

```
  choice = sample(1:5,1,prob=prob[[1]])
  weather24_pred$prediction[1] = states[choice]

  for(i in 2:nrow(weather24_pred)){
    if(month(weather24_pred$Date[i]) %in% c(12,1,2)){
      state_bin = one_hot(as.data.table(factor(weather24_pred$prediction[i-1], levels = states)))
      prob[[i]] = t(winter_tm) %*% t(state_bin)
      choice = sample(1:5,1,prob=prob[[i]])
      weather24_pred$prediction[i] = states[choice]
    }else if(month(weather24_pred$Date[i]) %in% c(3,4,5)){
      state_bin = one_hot(as.data.table(factor(weather24_pred$prediction[i-1], levels = states)))
      prob[[i]] = t(spring_tm) %*% t(state_bin)
      choice = sample(1:5,1,prob=prob[[i]])
      weather24_pred$prediction[i] = states[choice]
    }else if(month(weather24_pred$Date[i]) %in% c(6,7,8)){
      state_bin = one_hot(as.data.table(factor(weather24_pred$prediction[i-1], levels = states)))
      prob[[i]] = t(summer_tm) %*% t(state_bin)
      choice = sample(1:5,1,prob=prob[[i]])
      weather24_pred$prediction[i] = states[choice]
    }else if(month(weather24_pred$Date[i]) %in% c(9,10,11)){
      state_bin = one_hot(as.data.table(factor(weather24_pred$prediction[i-1], levels = states)))
      prob[[i]] = t(fall_tm) %*% t(state_bin)
      choice = sample(1:5,1,prob=prob[[i]])
      weather24_pred$prediction[i] = states[choice]
    }
  }
  return(weather24_pred)
}


seasonal_forecast <- function(forecast_df){
  prob = list()

  # First entry
  prob[[1]] = t(spring_tm) %*% c(0,0,0,0,1)
  choice = sample(1:5,1,prob=prob[[1]])
  forecast_df$prediction[1] = states[choice]

  for(i in 2:nrow(forecast_df)){
    if(month(forecast_df$Date[i]) %in% c(12,1,2)){
      state_bin = one_hot(as.data.table(factor(forecast_df$prediction[i-1], levels = states)))
      prob[[i]] = t(winter_tm) %*% t(state_bin)
      choice = sample(1:5,1,prob=prob[[i]])
      forecast_df$prediction[i] = states[choice]
    }else if(month(forecast_df$Date[i]) %in% c(3,4,5)){
      state_bin = one_hot(as.data.table(factor(forecast_df$prediction[i-1], levels = states)))
      prob[[i]] = t(spring_tm) %*% t(state_bin)
      choice = sample(1:5,1,prob=prob[[i]])
      forecast_df$prediction[i] = states[choice]
    }else if(month(forecast_df$Date[i]) %in% c(6,7,8)){
      state_bin = one_hot(as.data.table(factor(forecast_df$prediction[i-1], levels = states)))
      prob[[i]] = t(summer_tm) %*% t(state_bin)
      choice = sample(1:5,1,prob=prob[[i]])
```

```
      forecast_df$prediction[i] = states[choice]
    }else if(month(forecast_df$Date[i]) %in% c(9,10,11)){
      state_bin = one_hot(as.data.table(factor(forecast_df$prediction[i-1], levels = states)))
      prob[[i]] = t(fall_tm) %*% t(state_bin)
      choice = sample(1:5,1,prob=prob[[i]])
      forecast_df$prediction[i] = states[choice]
    }
  }
  return(forecast_df)
}
```

```
S = 50# Number of Simulations
someData <- rep(NA, 4*5*S)  #(Months, States, S)
avg_state <- array(someData, c(4, 5, S))

for(s in 1:S){
  weather24_pred$prediction<- rep(NA,nrow(weather24_pred))
  temp = seasonal(weather24_pred)
  temp_avg = temp %>% group_by(Month) %>% mutate(month_len = length(Month)) %>%
    ungroup() %>%
    group_by(Month, prediction) %>%
    reframe(ratio = length(prediction)/month_len)%>%
    unique()
  for(i in 1:nrow(temp_avg)){
    if(temp_avg$Month[i]==1){
      if(temp_avg$prediction[i]=="Chilly"){
        avg_state[1,1,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Cloudy"){
        avg_state[1,2,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Partly Cloudy"){
        avg_state[1,3,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Rainy"){
        avg_state[1,4,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Sunny"){
        avg_state[1,5,s] = temp_avg$ratio[i]
      }
    }else if(temp_avg$Month[i]==2){
      if(temp_avg$prediction[i]=="Chilly"){
        avg_state[2,1,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Cloudy"){
        avg_state[2,2,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Partly Cloudy"){
        avg_state[2,3,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Rainy"){
        avg_state[2,4,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Sunny"){
        avg_state[2,5,s] = temp_avg$ratio[i]
      }
    }else if(temp_avg$Month[i]==3){
      if(temp_avg$prediction[i]=="Chilly"){
        avg_state[3,1,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Cloudy"){
        avg_state[3,2,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Partly Cloudy"){
```

```
        avg_state[3,3,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Rainy"){
        avg_state[3,4,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Sunny"){
        avg_state[3,5,s] = temp_avg$ratio[i]
      }
    }else if(temp_avg$Month[i]==4){
      if(temp_avg$prediction[i]=="Chilly"){
        avg_state[4,1,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Cloudy"){
        avg_state[4,2,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Partly Cloudy"){
        avg_state[4,3,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Rainy"){
        avg_state[4,4,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Sunny"){
        avg_state[4,5,s] = temp_avg$ratio[i]
      }
    }
  }
  avg_state[is.na(avg_state)] <- 0
  }
```

```
someData <- rep(0, 4*5)
sum_mat = array(someData, c(4, 5))
for(s in 1:S){
  sum_mat = sum_mat + avg_state[,,s]
}

sum_mat = sum_mat/S
round(sum_mat,2)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 0.51 0.30 0.02 0.11 0.06
## [2,] 0.54 0.27 0.01 0.10 0.07
## [3,] 0.50 0.31 0.00 0.05 0.13
## [4,] 0.52 0.29 0.00 0.04 0.14
```

```
mse<-function(x_hat,x) rowMeans((x_hat-x)^2)

weather_ratio = matrix(c(0.58064516, 0.25806452, 0.06451613, 0.06451613,0.03225806,
                         0.41379310, 0.34482759, 0, 0.24137931,0,
                         0.54838710, 0.29032258, 0, 0.12903226, 0.03225806,
                         0.6, 0.33333333, 0, 0.03333333, 0.03333333), nrow = 4, ncol = 5, byrow = TRUE)
sum(mse(sum_mat, weather_ratio))
```

```
## [1] 0.01928177
```

Forecasting

```
forecast_df = data.frame(Date=rep(seq(as.Date('2024-05-01'),
                                  as.Date('2024-12-31'),
                                  by = 'days')))
forecast_df %<>% mutate(Month= month(Date),
```

```
                        prediction = rep(NA,nrow(forecast_df)))
```

```
S = 50# Number of Simulations
someData <- rep(NA, 12*5*S)  #(Months, States, S)
avg_state_pred <- array(someData, c(12, 5, S)) # Initializing 3D matrix

for(s in 1:S){
  forecast_df$prediction<- rep(NA,nrow(forecast_df))
  temp = seasonal_forecast(forecast_df)
  temp_avg = temp %>% group_by(Month) %>% mutate(month_len = length(Month)) %>%
    ungroup() %>%
    group_by(Month, prediction) %>%
    reframe(ratio = length(prediction)/month_len)%>%
    unique()
  for(i in 1:nrow(temp_avg)){
    if(temp_avg$Month[i]==5){
      if(temp_avg$prediction[i]=="Chilly"){
        avg_state_pred[5,1,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Cloudy"){
        avg_state_pred[5,2,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Partly Cloudy"){
        avg_state_pred[5,3,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Rainy"){
        avg_state_pred[5,4,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Sunny"){
        avg_state_pred[5,5,s] = temp_avg$ratio[i]
      }
    }else if(temp_avg$Month[i]==6){
      if(temp_avg$prediction[i]=="Chilly"){
        avg_state_pred[6,1,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Cloudy"){
        avg_state_pred[6,2,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Partly Cloudy"){
        avg_state_pred[6,3,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Rainy"){
        avg_state_pred[6,4,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Sunny"){
        avg_state_pred[6,5,s] = temp_avg$ratio[i]
      }
    }else if(temp_avg$Month[i]==7){
      if(temp_avg$prediction[i]=="Chilly"){
        avg_state_pred[7,1,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Cloudy"){
        avg_state_pred[7,2,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Partly Cloudy"){
        avg_state_pred[7,3,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Rainy"){
        avg_state_pred[7,4,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Sunny"){
        avg_state_pred[7,5,s] = temp_avg$ratio[i]
      }
    }else if(temp_avg$Month[i]==8){
      if(temp_avg$prediction[i]=="Chilly"){
        avg_state_pred[8,1,s] = temp_avg$ratio[i]
```

```
    }else if(temp_avg$prediction[i]=="Cloudy"){
      avg_state_pred[8,2,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Partly Cloudy"){
      avg_state_pred[8,3,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Rainy"){
      avg_state_pred[8,4,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Sunny"){
      avg_state_pred[8,5,s] = temp_avg$ratio[i]
    }
  }else if(temp_avg$Month[i]==9){
    if(temp_avg$prediction[i]=="Chilly"){
      avg_state_pred[9,1,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Cloudy"){
      avg_state_pred[9,2,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Partly Cloudy"){
      avg_state_pred[9,3,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Rainy"){
      avg_state_pred[9,4,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Sunny"){
      avg_state_pred[9,5,s] = temp_avg$ratio[i]
    }
  }else if(temp_avg$Month[i]==10){
    if(temp_avg$prediction[i]=="Chilly"){
      avg_state_pred[10,1,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Cloudy"){
      avg_state_pred[10,2,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Partly Cloudy"){
      avg_state_pred[10,3,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Rainy"){
      avg_state_pred[10,4,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Sunny"){
      avg_state_pred[10,5,s] = temp_avg$ratio[i]
    }
  }else if(temp_avg$Month[i]==11){
    if(temp_avg$prediction[i]=="Chilly"){
      avg_state_pred[11,1,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Cloudy"){
      avg_state_pred[11,2,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Partly Cloudy"){
      avg_state_pred[11,3,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Rainy"){
      avg_state_pred[11,4,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Sunny"){
      avg_state_pred[11,5,s] = temp_avg$ratio[i]
    }
  }else if(temp_avg$Month[i]==12){
    if(temp_avg$prediction[i]=="Chilly"){
      avg_state_pred[12,1,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Cloudy"){
      avg_state_pred[12,2,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Partly Cloudy"){
      avg_state_pred[12,3,s] = temp_avg$ratio[i]
    }else if(temp_avg$prediction[i]=="Rainy"){
```

```
        avg_state_pred[12,4,s] = temp_avg$ratio[i]
      }else if(temp_avg$prediction[i]=="Sunny"){
        avg_state_pred[12,5,s] = temp_avg$ratio[i]
      }
    }
  }
  avg_state_pred[is.na(avg_state_pred)] <- 0
}
```

```
someData <- rep(0, 12*5)
sum_mat = array(someData, c(12, 5))
for(s in 1:S){
  sum_mat = sum_mat + avg_state_pred[,,s]
}

sum_mat = sum_mat/S

round(sum_mat[c(5:12),],2)
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] 0.52 0.28 0.00 0.04 0.16
## [2,] 0.04 0.20 0.17 0.00 0.60
## [3,] 0.03 0.19 0.20 0.01 0.58
## [4,] 0.04 0.19 0.16 0.01 0.59
## [5,] 0.19 0.20 0.07 0.01 0.53
## [6,] 0.18 0.19 0.06 0.01 0.56
## [7,] 0.16 0.18 0.07 0.01 0.57
## [8,] 0.54 0.27 0.02 0.09 0.09
```