SDM

Koissi Savi

2024-04-24

Exploratory data analysis

```
# Load necessary libraries
library(readxl)
library(sp)
library(sf)
```

```
## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```
#library(rgdal)
library(raster)
library(caret)
```

```
## Loading required package: ggplot2
```

```
## Loading required package: lattice
```

```
#install.packages("adehabitatHS")
library(adehabitatHS) # ENFA-SDM
```

```
## Loading required package: ade4
```

```
## Loading required package: adehabitatMA
```

```
## Registered S3 methods overwritten by 'adehabitatMA':
##   method                       from
##   print.SpatialPixelsDataFrame sp
##   print.SpatialPixels          sp
```

```
##
## Attaching package: 'adehabitatMA'
```

```
## The following object is masked from 'package:raster':
##
##     buffer
```

```
## Loading required package: adehabitatHR
```

```
## Loading required package: adehabitatLT
```

```
## Loading required package: CircStats
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following objects are masked from 'package:raster':
##
```

```
##     area, select
```

```
## Loading required package: boot
```

```
##
## Attaching package: ’boot’
```

```
## The following object is masked from ’package:lattice’:
##
##     melanoma
```

```r
library(tidyverse)
```

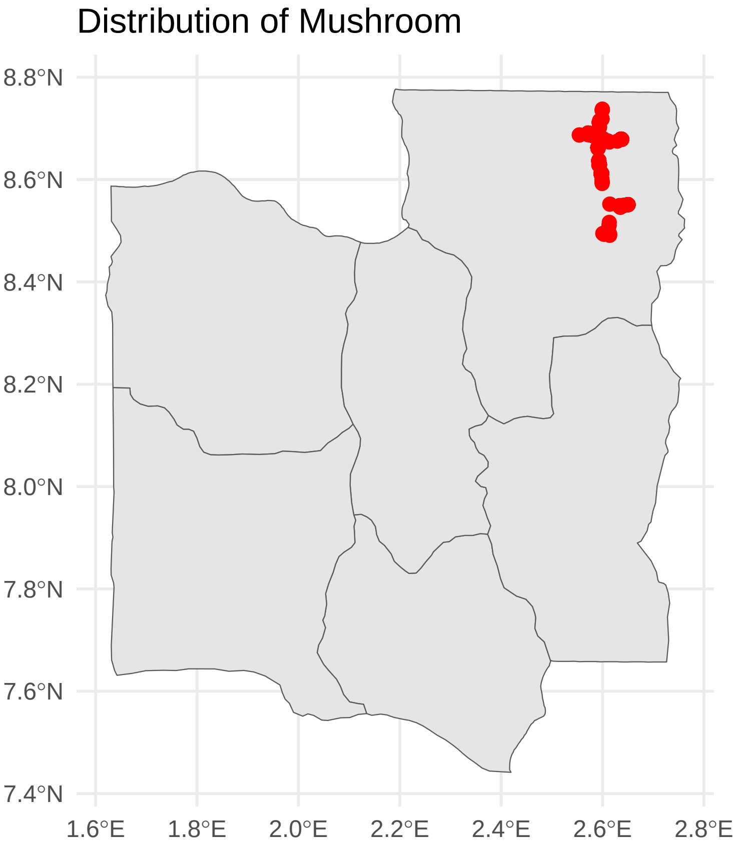
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::extract() masks raster::extract()
## x dplyr::filter()  masks stats::filter()
## x dplyr::id()      masks adehabitatLT::id()
## x dplyr::lag()     masks stats::lag()
## x purrr::lift()    masks caret::lift()
## x dplyr::select()  masks MASS::select(), raster::select()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
# EXPLORATORY DATA ANALYSIS
data <- read_rds("mash.RDS") # Read dataset

# Visualize on a map
# Define the coordinate reference system (CRS) for latitude and longitude data
latlong_crs <- CRS("+proj=longlat +datum=WGS84")

# Convert to spatial data
spatial_points <- data %>%
  dplyr::select(longitude, latitude) %>%
  st_as_sf(coords = c("longitude", "latitude"), crs = latlong_crs)

# Define the desired projection for your map
map_projection <- "+proj=merc +ellps=WGS84"

# Convert sf object to sp object
spatial_points_sp <- as(spatial_points, "Spatial")

# Transform spatial points using map projection
projected_points <- st_transform(spatial_points, map_projection)

# Load your shapefile
background_shapefile <- st_read("Ben_shapefile/BEN_adm2.shp")
```

```
## Reading layer ‘BEN_adm2’ from data source
##   ‘/Users/koissi/Desktop/SDM/sdm/Ben_shapefile/BEN_adm2.shp’
##   using driver ‘ESRI Shapefile’
## Simple feature collection with 76 features and 18 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 0.774574 ymin: 6.23514 xmax: 3.851701 ymax: 12.41835
## Geodetic CRS:  WGS 84
```

```
background_shapefile_F <- background_shapefile %>%
  filter(NAME_1 == "Collines")

# Plot the background shapefile with projected points
# Calculate species counts

ggplot() +
  geom_sf(data = background_shapefile_F) +
  geom_sf(data = projected_points, color = "red", size = 2) +
  labs(title = "Distribution of Mushroom") +
  theme_minimal()
```



# Profile Technique

The three methods described here, Bioclim, Domain, and Mahal. These methods are implemented in the dismo package, and the procedures to use these models are the same for all three.

Preparation of the dataset

Let us recreate the data we have used so far.

```
# Step 1: Load required packages
library(dismo)
```

```
##
## Attaching package: 'dismo'

## The following object is masked from 'package:adehabitatHS':
```

```
##
##     domain
```

```
#library(maptools)

# EXPLORATORY DATA ANALYSIS
data <- read_rds("mash.RDS") # Read dataset

# Assuming presence data can be derived from the 'Espèce' column
presence_species <- unique(data$Espèce)

directories <-c("data_Connexe/", "data_Connexe/wc2.1_10m_tmax/",  "data_Connexe/wc2.1_10m_tavg/")

# List files from each directory and combine them into a single vector
predictor_files <- unlist(lapply(directories, function(dir) {
  list.files(dir, pattern = '.tif$', full.names = TRUE)
}))

# Stack all predictor files
predictors <- stack(predictor_files)

# Extract presence and background points
presence_points <- data %>%
  dplyr::filter(Espèce %in% presence_species) %>%
  st_as_sf(coords = c("longitude", "latitude"), crs = latlong_crs)

set.seed(123)


background_points <- randomPoints(predictors, 500)

# Extract environmental values at presence and background points
presvals <- raster::extract(predictors, presence_points)
absvals <- raster::extract(predictors, background_points)

# Create a binary response variable indicating presence (1) or absence (0)
pb <- c(rep(1, nrow(presvals)), rep(0, nrow(absvals)))

# Combine presence and absence data
sdmdata <- data.frame(presence = c(rep(1, length(presvals)), rep(0, length(absvals))),
                      rbind(presvals, absvals))

# Rename the columns
names(sdmdata) <- c("presence", names(sdmdata)[-1])

# View the resulting data frame
print(sdmdata %>% head())
```

```
##   presence wc2.1_10m_elev.2 wc2.1_10m_tmax_01 wc2.1_10m_tmax_02
## 1        1              282           34.4835          36.03225
## 2        1              282           34.4835          36.03225
## 3        1              282           34.4835          36.03225
## 4        1              282           34.4835          36.03225
## 5        1              282           34.4835          36.03225
## 6        1              282           34.4835          36.03225
##   wc2.1_10m_tmax_03 wc2.1_10m_tmax_04 wc2.1_10m_tmax_05 wc2.1_10m_tmax_06
## 1          35.68075            34.024          32.45575          30.64575
## 2          35.68075            34.024          32.45575          30.64575
## 3          35.68075            34.024          32.45575          30.64575
## 4          35.68075            34.024          32.45575          30.64575
## 5          35.68075            34.024          32.45575          30.64575
## 6          35.68075            34.024          32.45575          30.64575
##   wc2.1_10m_tmax_07 wc2.1_10m_tmax_08 wc2.1_10m_tmax_09 wc2.1_10m_tmax_10
## 1            28.681            28.019          29.24125            30.753
## 2            28.681            28.019          29.24125            30.753
## 3            28.681            28.019          29.24125            30.753
## 4            28.681            28.019          29.24125            30.753
## 5            28.681            28.019          29.24125            30.753
## 6            28.681            28.019          29.24125            30.753
##   wc2.1_10m_tmax_11 wc2.1_10m_tmax_12 wc2.1_10m_tavg_01 wc2.1_10m_tavg_02
## 1          33.30175            33.696          27.01375           28.6335
## 2          33.30175            33.696          27.01375           28.6335
## 3          33.30175            33.696          27.01375           28.6335
## 4          33.30175            33.696          27.01375           28.6335
## 5          33.30175            33.696          27.01375           28.6335
## 6          33.30175            33.696          27.01375           28.6335
##   wc2.1_10m_tavg_03 wc2.1_10m_tavg_04 wc2.1_10m_tavg_05 wc2.1_10m_tavg_06
## 1            29.145          28.33775           27.3045          26.02325
## 2            29.145          28.33775           27.3045          26.02325
## 3            29.145          28.33775           27.3045          26.02325
## 4            29.145          28.33775           27.3045          26.02325
## 5            29.145          28.33775           27.3045          26.02325
## 6            29.145          28.33775           27.3045          26.02325
##   wc2.1_10m_tavg_07 wc2.1_10m_tavg_08 wc2.1_10m_tavg_09 wc2.1_10m_tavg_10
## 1          24.83875           24.3225          25.07225           25.8995
## 2          24.83875           24.3225          25.07225           25.8995
## 3          24.83875           24.3225          25.07225           25.8995
## 4          24.83875           24.3225          25.07225           25.8995
## 5          24.83875           24.3225          25.07225           25.8995
## 6          24.83875           24.3225          25.07225           25.8995
##   wc2.1_10m_tavg_11 wc2.1_10m_tavg_12
## 1            26.821          26.40125
## 2            26.821          26.40125
## 3            26.821          26.40125
## 4            26.821          26.40125
## 5            26.821          26.40125
## 6            26.821          26.40125
```

```r
# Extract coordinates
# Extract coordinates from 'data'
existing_coords <- data[, c("longitude", "latitude")]

# Extract coordinates from 'background_points'
background_coords <- coordinates(background_points)

# Rename the columns of 'background_coords' to match 'existing_coords'
colnames(background_coords) <- c("longitude", "latitude")

# Combine existing coordinates with random background points
```

```r
all_coords <- rbind(existing_coords, background_coords)

# Create 'loc' object
loc <- data.frame(longitude = all_coords[, 1], latitude = all_coords[, 2])
```

Split the data into a training set and a testing set

```r
set.seed(457)
group <- kfold(sdmdata , 5)
pres_train <- sdmdata [group != 1, ]
pres_test <- sdmdata [group == 1, ]
```

Bioclim

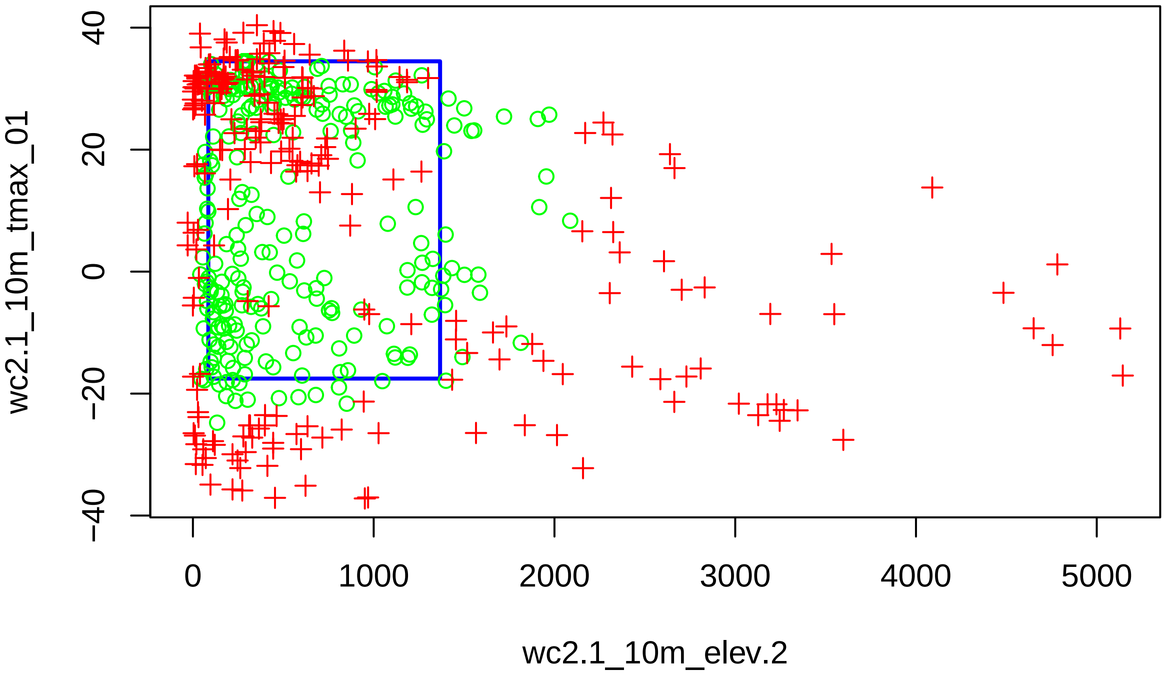
Modeling the dispersal of species has made substantial use of the BIOCLIM algorithm. A traditional "climate-envelope-model" is BIOCLIM (Booth et al., 2014). Despite its lower performance compared to other modeling techniques (Elith et al., 2006), especially when considering climate change (Hijmans and Graham, 2006), the algorithm's ease of comprehension makes it a valuable tool for teaching species distribution modeling, which is one of the reasons it is still in use. By comparing the values of environmental variables at any location to a percentile distribution of the values at known locations of occurrence (referred to as "training sites"), the BIOCLIM algorithm calculates the similarity of a location. The placement is more appropriate the closer it is to the median, or the 50th percentile.

The distribution's tails are considered equally, meaning that the 10 percentile and the 90 percentile are the same. The least percentile score across all environmental variables is employed in the "dismo" implementation, which converts upper tail numbers to lower tail values (i.e., BIOCLIM adopts a technique similar to Liebig's law of the minimum). To get a result between 0 and 1, this value is deducted from 1 and then multiplied by 2. Scaling in this manner makes the results easier to interpret by becoming more similar to those of other distribution modeling techniques.

Rarely will the value 1 be detected because it would need to be at a position where the training data's median value for each variable taken into account is present. Since it is applied to every cell whose value of an environmental variable falls outside the range of the training data, or the percentile distribution, for at least one of the variables, the value 0 is extremely common.

Previously, we used data.frame to fit a Bioclim model, where each row represented environmental data at known places of a species' existence. Here, we only need to use the predictors and the occurrence points to build a bioclimatic model—the function will take care of the extraction.

```r
bc <- bioclim(predictors, loc)
plot(bc, a=1, b=2, p=0.85)
```



Domain

For modeling species distribution, the Domain algorithm (Carpenter et al., 1993) has been widely utilized. It fared poorly when evaluating the effects of climate change (Hijmans and Graham, 2006) and poorly in a model comparison (Elith et al., 2006). The Gower distance between environmental variables at any given location and those at any known locations of occurrence (sometimes referred to as "training sites") is calculated by the Domain algorithm.

For each climate variable, the distance between the known occurrences and the environment at point A is determined by dividing the absolute difference in the variable's values across all known occurrence points by the variable's range (i.e., the distance is scaled by the range of observations). The shortest path between a site and any training point is measured for each variable. The average of these distances across all environmental factors is then the Gower distance. The algorithm determines a location's distance (in environmental space) from the nearest known occurrence.

The distance to any variable is used to integrate over environmental variables. To get the scores between 0 (low) and 1 (high), this distance is deducted from one, and (in this R version) numbers below zero are truncated.

Below we fit a domain model, evaluate it, and make a prediction. We map the prediction, as well as a map subjectively classified into presence / absence.

```
dm <- domain(predictors, loc)
#pd = predict(predictors, dm,  progress='')
```

Regression-based models

Generalized Linear Models

A generalized linear model (GLM) is a generalization of ordinary least squares regression. Models are fit using maximum likelihood and by allowing the linear model to be related to the response variable via a link function and by allowing the magnitude of the variance of each measurement to be a function of its predicted value. Depending on how a GLM is specified it can be equivalent to (multiple) linear regression, logistic regression or Poisson regression. See Guisan et al (2002) for an overview of the use of GLM in species distribution modeling.

```
# logistic regression:
gm1 <- glm(presence ~ wc2.1_10m_elev.2 + wc2.1_10m_tmax_01 + wc2.1_10m_tmax_02 +
             wc2.1_10m_tmax_03 + wc2.1_10m_tmax_04 + wc2.1_10m_tmax_05 +
             wc2.1_10m_tmax_06 + wc2.1_10m_tmax_07 +  wc2.1_10m_tmax_08 +
             wc2.1_10m_tmax_09 + wc2.1_10m_tmax_10 +
             wc2.1_10m_tmax_11 + wc2.1_10m_tmax_12,
            family = binomial(link = "logit"), data=sdmdata)
summary(gm1)
```

```
##
## Call:
## glm(formula = presence ~ wc2.1_10m_elev.2 + wc2.1_10m_tmax_01 +
##     wc2.1_10m_tmax_02 + wc2.1_10m_tmax_03 + wc2.1_10m_tmax_04 +
##     wc2.1_10m_tmax_05 + wc2.1_10m_tmax_06 + wc2.1_10m_tmax_07 +
##     wc2.1_10m_tmax_08 + wc2.1_10m_tmax_09 + wc2.1_10m_tmax_10 +
##     wc2.1_10m_tmax_11 + wc2.1_10m_tmax_12, family = binomial(link = "logit"),
##     data = sdmdata)
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -6.080e-02  5.785e-02  -1.051    0.293
## wc2.1_10m_elev.2  -5.462e-06  2.508e-05  -0.218    0.828
## wc2.1_10m_tmax_01 -8.527e-03  3.038e-02  -0.281    0.779
## wc2.1_10m_tmax_02  1.052e-02  2.718e-02   0.387    0.699
## wc2.1_10m_tmax_03 -8.752e-04  2.642e-02  -0.033    0.974
## wc2.1_10m_tmax_04 -2.919e-03  3.077e-02  -0.095    0.924
## wc2.1_10m_tmax_05  2.257e-03  3.402e-02   0.066    0.947
## wc2.1_10m_tmax_06  2.389e-03  3.533e-02   0.068    0.946
## wc2.1_10m_tmax_07 -1.569e-03  3.894e-02  -0.040    0.968
## wc2.1_10m_tmax_08 -3.433e-03  3.543e-02  -0.097    0.923
## wc2.1_10m_tmax_09  1.170e-02  3.558e-02   0.329    0.742
## wc2.1_10m_tmax_10 -2.108e-02  3.274e-02  -0.644    0.520
## wc2.1_10m_tmax_11  1.462e-02  2.745e-02   0.533    0.594
## wc2.1_10m_tmax_12 -3.188e-03  3.049e-02  -0.105    0.917
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 33739  on 24349  degrees of freedom
## Residual deviance: 33735  on 24336  degrees of freedom
## AIC: 33763
##
## Number of Fisher Scoring iterations: 3
```

Machine Learning Methods

Flexible regression models that are non-parametric are those used in machine learning. Support vector machines, Random Forests, boosted regression trees, and artificial neural networks (ANN) are some of the techniques used. You can also utilize the Maxent software, which implements the most popular method (maxent) in species distribution modeling, through the dismo package. A clear overview of machine learning and its differences from “classical statistics” (model-based probabilistic inference) is given by Breiman (2001a). Hastie et al. (2009) offer arguably the most comprehensive summary of these techniques.

Maxent

MaxEnt (short for “Maximum Entropy”; Phillips et al., 2006) is the most widely used SDM algorithm. Elith et al. (2010) provide an explanation of the algorithm (and software) geared towards ecologists. MaxEnt is available as a stand-alone Java program. Dismo has a function ‘maxent’ that communicates with this program.

Because MaxEnt is implemented in dismo you can fit it like the profile methods (e.g. Bioclim). That is, you can provide presence points and a RasterStack. However, you can also first fit a model, like with the other methods such as glm. But in the case of MaxEnt you cannot use the formula notation.

```
#maxent()
## Loading required namespace: rJava
## This is MaxEnt version 3.4.3
#xm <- maxent(predictors, loc) # This approaching is crashing R
## This is MaxEnt version 3.4.3
#plot(xm)
```

```
library(maxnet)
library(mecofun)
set.seed(124)
# First, we randomly select 70% of the rows that will be used as training data
train_i <- sample(seq_len(nrow(sdmdata)), size=round(0.7*nrow(sdmdata)))

# Then, we can subset the training and testing data

sp_train <- sdmdata[train_i,]
sp_test <- sdmdata[-train_i,]
my_preds <- sdmdata %>%
  select(-presence)
# Fit Maxent
m_maxent <- maxnet(p=sp_train$presence, data=sp_train[, -1],
    maxnet.formula(p = sp_train$presence, data = sp_train[, -1], classes = "lh"))


pkr <-  c('wc2.1_10m_elev.2',
          sprintf('wc2.1_10m_tmax_%02d', 1:12),
          sprintf('wc2.1_10m_tavg_%02d', 1:12))

 #partial_response(m_maxent,sp_train[,pkr], main='Maxent',
 #               ylab='Occurrence probability')
```

Interpretation

This output is from the *glmnet* function, which fits regularized generalized linear models (GLMs) via penalized maximum likelihood. Here’s how to interpret the output:

- Df: This column represents the degrees of freedom of the model. In the context of penalized regression, such as ridge (L2) or lasso (L1) regression, the degrees of freedom are a measure of model complexity, indicating the number of non-zero coefficients in the model.
- %Dev: This column represents the percentage of deviance explained by the model. Deviance is a measure of model fit in generalized linear models, similar to the residual sum of squares in linear regression. A lower deviance indicates a better fit to the data.
- Lambda: Lambda is the penalty parameter that controls the amount of regularization applied to the model. In glmnet, a sequence of lambda values is typically provided, and the model is fit for each lambda value. Lambda is inversely related to the strength of regularization; larger values of lambda

correspond to stronger regularization.

In the output you provided, it seems that the model with all predictors (Df = 0) achieves perfect fit to the data (%Dev = 0) for a wide range of lambda values. This could indicate either overfitting or perfect separation of the data, depending on the context of your analysis and the nature of your data.

```r
# Performance measures of Maxent
(perf_maxent <- evalSDM(sp_test$presence, predict(m_maxent, sp_test[,pkr],  type='logistic') ))
```

```
##         AUC         TSS       Kappa      Sens      Spec       PCC            D2
## 1 0.5007174 0.006115645 0.006110346 0.5075843 0.4985314 0.5029432 -0.0008689886
##   thresh
## 1    0.5
```

Interpretation

- TSS (True Skill Statistic): TSS is a measure of the discriminatory power of the model. It ranges from -1 to 1, where higher values indicate better discrimination between presence and absence locations. A TSS of 1 indicates perfect discrimination, while 0 indicates no skill beyond random chance.
- Kappa: Kappa is a measure of the agreement between the observed and predicted classifications, adjusted for the agreement expected by chance. It ranges from -1 to 1, where values closer to 1 indicate better agreement between observed and predicted values.
- Sens (Sensitivity): Sensitivity, also known as true positive rate, measures the proportion of actual presence locations that are correctly classified as presence by the model. Higher values indicate better ability to correctly predict presence.
- Spec (Specificity): Specificity measures the proportion of actual absence locations that are correctly classified as absence by the model. Higher values indicate better ability to correctly predict absence.
- PCC (Pearson Correlation Coefficient): PCC measures the linear correlation between observed and predicted values. It ranges from -1 to 1, where 1 indicates perfect positive correlation, -1 indicates perfect negative correlation, and 0 indicates no correlation.

In your provided values:

TSS is approximately 0.5024, indicating moderate discriminatory power. Kappa is approximately 0.0019, suggesting poor agreement beyond chance. Sensitivity is approximately 0.5839, indicating that about 58.39% of actual presence locations are correctly classified. Specificity is approximately 0.4181, indicating that about 41.81% of actual absence locations are correctly classified. PCC is approximately 0.4999, suggesting moderate positive correlation between observed and predicted values.

Overall, these metrics suggest that the model has some discriminatory power but performs poorly in terms of agreement and specificity. Further evaluation and possibly model refinement may be necessary to improve its performance.