FSVM: Training Models

SDMs & Machine Learning

Robert Ritson, WMI Research Associate

2021-08-05

Training Species Distributions Models with Machine Learning

The purpose of this vignette is to illustrate the process of training species distribution models with machine learning with `fsvm` using IDFG vegetation field data (already formatted and prepared for analysis). Please see "Data Preparation" and "Extract QuadPolyID" vignettes for details on formatting field data. This workflow includes accessing vegetation data from the IFWIS SQL data base (`getSQLData`), accessing the eCognition polygon covariate data developed by the University of Idaho (`getCovariates`), creating an `fsvm` object for model training (`as_fsvm`), training machine learning models (`fsvm_train` and `fsvm_train_lite`), and iterating the process through multiple species in a loop (`getModels`).

Machine Learning SDM Algorithms

Below is the list of algorithms utilized by `fsvm_train` and `fsvm_train_lite` to model species distributions with the R package `caret`. Required additional packages not already loaded by `caret` have been included as dependencies in `fsvm`. The training functions attempt to fit every algorithm listed for a particular data type and then select the best model based on the largest AUC value. If using `fsvm_train`, you will have to load the packages `nodeHarvest`, `bartMachine`, and `HDclassif` separately. Also be aware that `bartMachine` requires a valid installation of Java on your local machine.

Presence algorithms:

- Generalized linear model (GLM)
- Stochastic gradient boosting machine (GBM)
- Elastic-net regularized general linear model (GLMNET)
- Elastic-net regularized general linear model for classification (GLMNET_class)
- Random forest (RF)
- Extreme gradient boosting (XGBOOST)
- Self-organizing map (SOM)
- Bayesian generalized linear model (B_GLM)

- Support vector machine with polynomial kernel (SVM_P)
- Support vector machine with radial kernel (SVM_R)
- Linear discriminant analysis (LDA)
- Shrinkage discriminant analysis (SDA)
- Naive bayes classifier (NB_class)
- **Node harvest** (Tree_ensemble)
- **Bayesian additive regression tree** (BART)
- **Neural network** (NNET)
- **High-dimensional discriminant analysis** (HDDA)

Percent cover algorithms:

- Linear model (LM)
- Generalized linear model (GLM)
- Stochastic gradient boosting machine (GBM)
- Classification and Regression Trees (RPART)
- Classification and Regression Trees (RPAR2)
- Random forest (RF)
- Extreme gradient boosting (XGBOOST)
- Bagged additive regression tree (TREEBAG)
- Bagged multivariate adaptive regression spline (BAGEARTH)
- Self-organizing map (SOM)
- Bayesian generalized linear model (B_GLM)
- Support vector machine with polynomial kernel (SVM_P)
- Support vector machine with radial kernel (SVM_R)
- Neural network (NNET)
- Bayesian regularized neural network (BRNN)
- Dynamic evolving neural fuzzy inference system (DENFIS)
- Least absolute shrinkage and selection (LASSO)
- Bayesian least absolute shrinkage and selection (BLASSO)
- Bayesian ridge regression (BRIDGE)
- Genetic algorithm for tree modeling (EVTREE)
- **Node harvest** (Tree_ensemble)
- **Bayesian additive regression tree** (BART)
- **Model averaged neural network** (AVNNET)

Only available with 'fsvm_train'

Training Machine Learning Models

Install and load required packages

Install the latest version of `fsvm` and load required packages.

```
#Install latest version of `fsvm`
remotes::install_gitlab("idfg-r/fsvm_package", subdir = "pkg",auth_token = "oYfSyynwxTaobvGua9tF")
> Downloading GitLab repo idfg-r/fsvm_package@HEAD
> from URL https://gitlab.com/api/v4/projects/28272719/repository/archive.tar.gz?sha=HEAD
> Installing 1 packages: curl
> Installing package into 'C:/Users/rritson/Documents/R/win-library/4.0'
> (as 'lib' is unspecified)
> Installing package into 'C:/Users/rritson/Documents/R/win-library/4.0'
> (as 'lib' is unspecified)

#Load packages
lapply(c("fsvm","caret"),require,character.only=T)
> Loading required package: fsvm
> Loading required package: caret
> Loading required package: lattice
> Loading required package: ggplot2
```

Loading Data

Load formatted vegetation field data from the IFWIS SQL Database and covariates corresponding with the locations of species observations (by 'QuadPolyID'). `getSQLData` defaults to the sqltable 'Veg_fsvm_understory_model_data' which contains the formatted vegetation data, however, another valid IFWIS SQL data table could be specified to the function if necessary. `getCovariates` uses the default file path "A:/Fine scale vegetation analysis/fsvm_package/Covariates.RDS" to access query the .rds covariate files by 100k Quad polygon. "A" refers to the HQWILDSTAT network drive, if it is called something different on your machine, it must be explicitly specified in the 'cov.file.path' parameter. This function also defaults to rm.na=TRUE and export=FALSE which automatically removes NA records from the covariate file and refrains from saving the covariate file as an RData file to your working directory, both of which can be changed if necessary.

```
#Load field data
fielddata <- fsvm::getSQLData()
head(fielddata)
>   row_names       TranKey       PlotKey             Source DataType SampleYear
> 1         1  CPNWH_872726  CPNWH_872726 CPNWH_ID_Herbarium Presence       1970
> 2         2  CPNWH_878330  CPNWH_878330 CPNWH_ID_Herbarium Presence       1970
> 3         3  CPNWH_925921  CPNWH_925921 CPNWH_ID_Herbarium Presence       1932
> 4         4 CPNWH_3266691 CPNWH_3266691 CPNWH_ID_Herbarium Presence       2019
> 5         5 CPNWH_3266690 CPNWH_3266690 CPNWH_ID_Herbarium Presence       2019
> 6         6 CPNWH_3266688 CPNWH_3266688 CPNWH_ID_Herbarium Presence       2019
>   PercentCover PlotArea                        SpeciesName
> 1            0        0                Trillium petiolatum
> 2            0        0              Fritillaria lanceolata
> 3            0        0 Amelanchier alnifolia var. cusickii
> 4            0        0                Arceuthobium laricis
> 5            0        0                     Mycelis muralis
> 6            0        0                  Oxalis corniculata
>                       AcceptedName           G1                        G2   G3
> 1              Trillium petiolatum     Trillium       Trillium petiolatum <NA>
> 2 Fritillaria affinis var. affinis  Fritillaria       Fritillaria affinis <NA>
```

```
> 3             Amelanchier cusickii  Amelanchier      Amelanchier cusickii <NA>
> 4        Arceuthobium campylopodum Arceuthobium Arceuthobium campylopodum <NA>
> 5                  Mycelis muralis      Mycelis           Mycelis muralis <NA>
> 6              Oxalis corniculata       Oxalis        Oxalis corniculata <NA>
>                                 G4  G1TSN  G2TSN G3TSN  G4TSN TaxaKingdom
> 1                             <NA>  43054  43083  <NA>   <NA>     Plantae
> 2 Fritillaria affinis var. affinis  42932 507870  <NA> 531396     Plantae
> 3                             <NA>  25108 508697  <NA>   <NA>     Plantae
> 4                             <NA>  27886  27890  <NA>   <NA>     Plantae
> 5                             <NA> 500432 503893  <NA>   <NA>     Plantae
> 6                             <NA>  29062  29067  <NA>   <NA>     Plantae
>     TaxaPhylum     TaxaClass  TaxaOrder    TaxaFamily
> 1 Tracheophyta Magnoliopsida   Liliales Melanthiaceae
> 2 Tracheophyta Magnoliopsida   Liliales     Liliaceae
> 3 Tracheophyta Magnoliopsida    Rosales      Rosaceae
> 4 Tracheophyta Magnoliopsida Santalales   Santalaceae
> 5 Tracheophyta Magnoliopsida  Asterales    Asteraceae
> 6 Tracheophyta Magnoliopsida Oxalidales   Oxalidaceae
>                                                                  CommonName
> 1                                                          Idaho trillium
> 2                                  Checker lily,Checker lily,Checker lily
> 3                                                 Cusick's serviceberry
> 4                                               Western dwarf mistletoe
> 5                                                          Wall-lettuce
> 6 'Ihi,Creeping oxalis,Yellow oxalis,Yellow wood sorrel,Creeping woodsorrel
>   TaxonID Elk Moose MuleDeer SageGrouse    QuadPolyID Easting Northing EcoCode
> 1   49076   N     N        N          N q47116d8_2146 2274771  1813965   M333A
> 2   87798   N     N        N          N q47116d8_2146 2274771  1813965   M333A
> 3      NA   Y     Y        Y          N q47116e7_1021 2284256  1828336   M333A
> 4   46958   N     N        N          N q47116e7_5205 2287516  1826851   M333A
> 5   61669   N     N        N          N q47116e7_5639 2287463  1826671   M333A
> 6   49055   N     N        N          N q47116e7_5645 2287575  1826634   M333A
>   ShapeLength ShapeArea Quad100k           TimeStamp
> 1        1134      4078 q47116d8 2021-07-30 20:33:42
> 2        1134      4078 q47116d8 2021-07-30 20:33:42
> 3        1260      2946 q47116e7 2021-07-30 20:33:42
> 4        1622      3279 q47116e7 2021-07-30 20:33:42
> 5        1452      3905 q47116e7 2021-07-30 20:33:42
> 6        1854      4653 q47116e7 2021-07-30 20:33:42

#Load covariate data
covariates <- fsvm::getCovariates(fielddata)
> Loading required package: dplyr
> Warning: package 'dplyr' was built under R version 4.0.5
>
> Attaching package: 'dplyr'
> The following objects are masked from 'package:stats':
>
>     filter, lag
> The following objects are masked from 'package:base':
>
>     intersect, setdiff, setequal, union
> Loading required package: foreach
```

```
> [1] "Processing: 1 % complete"
> [1] "Processing: 3 % complete"
> [1] "Processing: 4 % complete"
> [1] "Processing: 5 % complete"
> [1] "Processing: 7 % complete"
> [1] "Processing: 8 % complete"
> [1] "Processing: 9 % complete"
> [1] "Processing: 11 % complete"
> [1] "Processing: 12 % complete"
> [1] "Processing: 13 % complete"
> [1] "Processing: 15 % complete"
> [1] "Processing: 16 % complete"
> [1] "Processing: 17 % complete"
> [1] "Processing: 19 % complete"
> [1] "Processing: 20 % complete"
> [1] "Processing: 21 % complete"
> [1] "Processing: 23 % complete"
> [1] "Processing: 24 % complete"
> [1] "Processing: 25 % complete"
> [1] "Processing: 27 % complete"
> [1] "Processing: 28 % complete"
> [1] "Processing: 29 % complete"
> [1] "Processing: 31 % complete"
> [1] "Processing: 32 % complete"
> [1] "Processing: 33 % complete"
> [1] "Processing: 35 % complete"
> [1] "Processing: 36 % complete"
> [1] "Processing: 37 % complete"
> [1] "Processing: 39 % complete"
> [1] "Processing: 40 % complete"
> [1] "Processing: 41 % complete"
> [1] "Processing: 43 % complete"
> [1] "Processing: 44 % complete"
> [1] "Processing: 45 % complete"
> [1] "Processing: 47 % complete"
> [1] "Processing: 48 % complete"
> [1] "Processing: 49 % complete"
> [1] "Processing: 51 % complete"
> [1] "Processing: 52 % complete"
> [1] "Processing: 53 % complete"
> [1] "Processing: 55 % complete"
> [1] "Processing: 56 % complete"
> [1] "Processing: 57 % complete"
> [1] "Processing: 59 % complete"
> [1] "Processing: 60 % complete"
> [1] "Processing: 61 % complete"
> [1] "Processing: 63 % complete"
> [1] "Processing: 64 % complete"
> [1] "Processing: 65 % complete"
> [1] "Processing: 67 % complete"
> [1] "Processing: 68 % complete"
> [1] "Processing: 69 % complete"
> [1] "Processing: 71 % complete"
```

```
> [1] "Processing: 72 % complete"
> [1] "Processing: 73 % complete"
> [1] "Processing: 75 % complete"
> [1] "Processing: 76 % complete"
> [1] "Processing: 77 % complete"
> [1] "Processing: 79 % complete"
> [1] "Processing: 80 % complete"
> [1] "Processing: 81 % complete"
> [1] "Processing: 83 % complete"
> [1] "Processing: 84 % complete"
> [1] "Processing: 85 % complete"
> [1] "Processing: 87 % complete"
> [1] "Processing: 88 % complete"
> [1] "Processing: 89 % complete"
> [1] "Processing: 91 % complete"
> [1] "Processing: 92 % complete"
> [1] "Processing: 93 % complete"
> [1] "Processing: 95 % complete"
> [1] "Processing: 96 % complete"
> [1] "Processing: 97 % complete"
> [1] "Processing: 99 % complete"
> [1] "Processing: 100 % complete"
head(covariates)
>         ele        slp casp sasp twi lcv      sri tpi    minpr    maxpr
> 1: 1647.818  2.9109755   -9    8   8   0 688641.1   0 17.30515 53.29431
> 2: 1746.134  2.4073910    8   -5   7   0 685014.0   0 17.46965 60.83045
> 3: 1737.899  2.6872969    8    4  11   0 683328.2   0 17.53094 61.02185
> 4: 1684.023  0.9421845    9    0   8   0 685372.5   0 17.95375 42.88982
> 5: 2137.924 12.1323848   -9    0   6   0 726431.4   0 19.32099 59.42374
> 6: 1518.398  5.9869054   -2   -9   6   0 670515.2   0 17.15069 45.61002
>        tapr     mintp    maxtp      aws     clay    sand     silt      cec
> 1: 305.1254 -8.016835 29.77664 4.750000  6.20000 64.1000 29.80000  6.70000
> 2: 334.5069 -9.164092 28.65485 3.250000 20.00000 42.1000 37.90000 14.00000
> 3: 337.0866 -9.237004 28.71037 5.000000 23.50000  9.4000 67.10000 17.50000
> 4: 286.1189 -8.916549 28.70079 2.950000 12.10000 36.8000 51.10000 10.40000
> 5: 462.0371 -7.093409 25.83119 1.519692 17.03112 45.5839 37.45855 16.30885
> 6: 298.4350 -9.472748 29.24017 4.673740  5.65252 60.5813 33.76618 10.96187
>          d2r ph       om     caco3 tsf ff tc sc nass dev     water_m2
> 1: 119.00000  7 3.000000  1.000000   0  0  0  0  152  52 0.000000e+00
> 2: 201.00000  7 2.000000  3.000000   0  0  0  0  176  81 0.000000e+00
> 3: 100.60145  7 3.500000  5.000000   0  0  0  0  176  81 0.000000e+00
> 4:  38.00000  8 0.840000 11.000000   0  0  0  1  152  71 6.981778e-05
> 5:  99.62173  7 3.159846  0.000000   0  0  0  3  152  52 1.414807e-05
> 6:  77.52520  8 2.961870  2.771221   0  0  0  0  152  81 0.000000e+00
>       shadow_m2 bareground_m2    mgrass_m2    xgrass_m2    mshrub_m2
> 1: 0.0000000000    0.81003584 0.0900836320 0.0900836320 0.0000000000
> 2: 0.0000000000    0.00000000 0.0000000000 0.0000000000 0.0000000000
> 3: 0.0000000000    0.00000000 0.0000000000 0.0000000000 0.0000000000
> 4: 0.0000000000    0.21064023 0.1434755289 0.1434755289 0.0002094533
> 5: 0.0001414807    0.03787439 0.0550925991 0.0550925991 0.0410860061
> 6: 0.0000000000    0.00000000 0.0007405213 0.0007405213 0.1972748815
>      xshrub_m2   conifer_m2     decid_m2 agriculture_m2 developed_m2 ytsf
> 1: 0.007407407 0.0009557945 0.0014336918      0.0000000            0 2021
```

```
> 2: 0.000000000 0.0000000000 0.0000000000      1.0000000            0 2021
> 3: 0.000000000 0.0000000000 0.0000000000      1.0000000            0 2021
> 4: 0.465684563 0.0363052433 0.0001396356      0.0000000            0 2021
> 5: 0.489792165 0.3119933221 0.0089132865      0.0000000            0 2021
> 6: 0.014218009 0.1295912322 0.0265106635      0.6309242            0 2021
>       ele2        slp2 casp2 twi2         sri2   minpr2   maxtp2      aws2
> 1: 2715304   8.4737783    81   64 474226626464 299.4684 886.6481 22.562500
> 2: 3048983   5.7955315    64   49 469244224359 305.1888 821.1006 10.562500
> 3: 3020292   7.2215648    64  121 466937365534 307.3338 824.2853 25.000000
> 4: 2835934   0.8877117    81   64 469735459407 322.3373 823.7356  8.702500
> 5: 4570719 147.1947614    81   36 527702583413 373.3007 667.2504  2.309462
> 6: 2305532  35.8430360     4   36 449590682200 294.1462 854.9875 21.843848
>        clay2      sand2    silt2     cec2      d2r2 ph2       om2     caco32
> 1:  38.44000 4108.80980  888.040  44.8900 14161.000  49  9.000000   1.000000
> 2: 400.00000 1772.40987 1436.410 196.0000 40401.000  49  4.000000   9.000000
> 3: 552.25000   88.35999 4502.410 306.2500 10120.652  49 12.250000  25.000000
> 4: 146.41001 1354.23994 2611.210 108.1600  1444.000  64  0.705600 121.000000
> 5: 290.05909 2077.89219 1403.143 265.9787  9924.489  49  9.984625   0.000000
> 6:  31.95098 3670.09383 1140.155 120.1626  6010.156  64  8.772675   7.679664
>      ytsf2 Real_Shape_Area     QuadPolyID Quad100k
> 1: 4084441            4185  q41113h2_5364 q41113h2
> 2: 4084441            2315   q41113h4_465 q41113h4
> 3: 4084441             690   q41113h4_527 q41113h4
> 4: 4084441           14323 q41113h5_10371 q41113h5
> 5: 4084441           70681   q41113h7_611 q41113h7
> 6: 4084441            6752   q41113h8_189 q41113h8
```

Format an `fsvm` modeling object (a species or group)

Now that you have the field data and covariate files loaded, you are ready create an `fsvm` modeling object which subsets observations of a specified species or group, divides the data by "presence" and "percent_cover" data types, and creates the appropriate response variable by QuadPolyID (either 1 or 0 for presence data or proportion for percent cover). The group parameter can be either 'G1', 'G2', 'G3', or 'G4' for genus, species, subspecies, or variety groupings. The parameter tax.list accepts the any label appropriate to the specified group. The resulting object contains 5 data frames: 'LpiCov', 'ObsData_plots', 'ObsData_pts', 'PLOTS', and 'PTS'. Only the 'PLOTS' and 'PTS' data frames are used by training functions for percent cover and presence modeling respectively. The first three data frames contain the original data and observation subsets.

```
#Select Group and Species - species, Woods rose
df.fsvm <- fsvm::as_fsvm(fielddata = fielddata, covariates = covariates,
                         group = "G2", tax.list = "Rosa woodsii")
> Joining, by = c("QuadPolyID", "Quad100k")
> Joining, by = c("QuadPolyID", "Quad100k")
> Joining, by = "QuadPolyID"
> Joining, by = "QuadPolyID"
head(df.fsvm$PTS) #data used for presence modeling
> # A tibble: 6 x 67
> # Groups:   QuadPolyID, ShapeLength, ShapeArea [6]
>   QuadPolyID ShapeLength ShapeArea EcoCode Total   Hit  Prop Present   ele   slp
>   <chr>            <dbl>     <dbl> <chr>   <int> <dbl> <dbl>   <dbl> <dbl> <dbl>
> 1 q42111b6_~         330.      580. M331D       7     0     0       0 1884.  28.8
> 2 q42111b6_~        2974.     9342. M331D      25     0     0       0 2122.  34.2
> 3 q42111b6_~        3520.    10418  M331D      18     0     0       0 2061.  30.7
> 4 q42111d8_~        1438.     5496. M331D      16     0     0       0 2054.  28.2
```

```
> 5 q42111d8_~        362.     1266. M331D      27     0     0       0 2026.  11.5
> 6 q42111d8_~       1000.     4210. M331D      45     0     0       0 2028.  14.6
> # ... with 57 more variables: casp <int>, sasp <int>, twi <int>, lcv <int>,
> #   sri <dbl>, tpi <fct>, minpr <dbl>, maxpr <dbl>, tapr <dbl>, mintp <dbl>,
> #   maxtp <dbl>, aws <dbl>, clay <dbl>, sand <dbl>, silt <dbl>, cec <dbl>,
> #   d2r <dbl>, ph <int>, om <dbl>, caco3 <dbl>, tsf <fct>, ff <fct>, tc <dbl>,
> #   sc <fct>, nass <fct>, dev <fct>, water_m2 <dbl>, shadow_m2 <dbl>,
> #   bareground_m2 <dbl>, mgrass_m2 <dbl>, xgrass_m2 <dbl>, mshrub_m2 <dbl>,
> #   xshrub_m2 <dbl>, conifer_m2 <dbl>, decid_m2 <dbl>, ...
head(df.fsvm$PLOTS) #data used for percent cover modeling
> # A tibble: 6 x 66
> # Groups:   QuadPolyID, ShapeLength, ShapeArea [6]
>   QuadPolyID     ShapeLength ShapeArea EcoCode Total  Prop Present   ele    slp
>   <chr>                <dbl>     <dbl> <chr>   <int> <dbl>   <dbl> <dbl>  <dbl>
> 1 q42111a1_10238      32384.   547458  -342E       9     0       0 2241. 10.6
> 2 q42111a1_7899        6406.    75771. -342E       9     0       0 2230.  7.47
> 3 q42111a1_9075        4180.    36322. -342E       9     0       0 2219. 12.2
> 4 q42111a7_20088        936.     5873. M331D      1     0       0 1367.  0.194
> 5 q42111a7_20361       1058.    14773  M331D      1     0       0 1367.  0.307
> 6 q42111a7_20417        442.     2331  M331D      1     0       0 1367.  0.382
> # ... with 57 more variables: casp <int>, sasp <int>, twi <int>, lcv <int>,
> #   sri <dbl>, tpi <fct>, minpr <dbl>, maxpr <dbl>, tapr <dbl>, mintp <dbl>,
> #   maxtp <dbl>, aws <dbl>, clay <dbl>, sand <dbl>, silt <dbl>, cec <dbl>,
> #   d2r <dbl>, ph <int>, om <dbl>, caco3 <dbl>, tsf <fct>, ff <fct>, tc <dbl>,
> #   sc <fct>, nass <fct>, dev <fct>, water_m2 <dbl>, shadow_m2 <dbl>,
> #   bareground_m2 <dbl>, mgrass_m2 <dbl>, xgrass_m2 <dbl>, mshrub_m2 <dbl>,
> #   xshrub_m2 <dbl>, conifer_m2 <dbl>, decid_m2 <dbl>, ...
```

Training species distribution models with machine learning (base function)

As previously mentioned, there are two base functions for training machine learning models, `fsvm_train` and `fsvm_train_lite`. Starting out, it is recommended to use `lite` as it is faster and less finicky. `fsvm_train` also includes model averaging algorithms which have yet to be fully evaluated. The two 'type' parameters are "presence" and "percent_cover", which selects which type of data is used to train the models, triggering the appropriate algorithm list (see 'Machine Learning SDM Algorithms' above). The functions return two objects, `fsvm.dat` which contains the data used to train the models and `fsvm.res` which contains all of the model training outputs including the best selected model.

```
###Train machine learning models (13 algorithms for presence data with lite)
df.train <- fsvm::fsvm_train_lite(DAT = df.fsvm, type = "presence") #this can take awhile...
> Setting levels: control = 0, case = 1
> Setting direction: controls < cases
> Setting levels: control = 0, case = 1
> Setting direction: controls < cases
> Setting levels: control = 0, case = 1
> Setting direction: controls < cases
> Setting levels: control = 0, case = 1
> Setting direction: controls < cases
> Setting levels: control = 0, case = 1
> Setting direction: controls < cases
> Setting levels: control = 0, case = 1
> Setting direction: controls < cases
> Setting levels: control = 0, case = 1
> Setting direction: controls < cases
```

```
> Setting levels: control = 0, case = 1
> Setting direction: controls < cases
> Setting levels: control = 0, case = 1
> Setting direction: controls < cases
> Setting levels: control = 0, case = 1
> Setting direction: controls < cases
> Setting levels: control = 0, case = 1
> Setting direction: controls < cases
> Setting levels: control = 0, case = 1
> Setting direction: controls < cases
> Setting levels: control = 0, case = 1
> Setting direction: controls < cases
```

```
head(df.train$DATA$train)  #training data
> # A tibble: 6 x 61
>   QuadPolyID  ShapeLength ShapeArea Total   Hit   Prop Present   ele   slp  casp
>   <chr>             <dbl>     <dbl> <int> <dbl>  <dbl>   <dbl> <dbl> <dbl> <int>
> 1 q42111b6_6~       2974.     9342.    25     0 0           0 2122.  34.2    -5
> 2 q42111d8_1~       1438.     5496.    16     0 0           0 2054.  28.2    -9
> 3 q42111d8_1~        362.     1266.    27     0 0           0 2026.  11.5    -9
> 4 q42111d8_1~       1000.     4210.    45     0 0           0 2028.  14.6     7
> 5 q42111d8_1~       2134.     5367.     7     0 0           0 2032.  17.2     7
> 6 q42111d8_1~       1648.     5796.    99     1 0.0101      1 2069.  20.1     7
> # ... with 51 more variables: sasp <int>, twi <int>, lcv <int>, sri <dbl>,
> #   minpr <dbl>, maxpr <dbl>, tapr <dbl>, mintp <dbl>, maxtp <dbl>, aws <dbl>,
> #   clay <dbl>, sand <dbl>, silt <dbl>, cec <dbl>, d2r <dbl>, ph <int>,
> #   om <dbl>, caco3 <dbl>, tc <dbl>, water_m2 <dbl>, shadow_m2 <dbl>,
> #   bareground_m2 <dbl>, mgrass_m2 <dbl>, xgrass_m2 <dbl>, mshrub_m2 <dbl>,
> #   xshrub_m2 <dbl>, conifer_m2 <dbl>, decid_m2 <dbl>, agriculture_m2 <dbl>,
> #   developed_m2 <dbl>, ytsf <int>, ele2 <dbl>, slp2 <dbl>, casp2 <dbl>, ...
df.train$BEST  #best model by max. AUC
> $max_ROC_caret
> [1] "RF"
>
> $min_ROCspread_caret
> [1] "SVM_P"
>
> $max_AUC_test
> [1] "RF"
>
> $max_AUC_test_val
> [1] 0.7790953
```

Iterating model training through multiple species (loop function)

For training machine learning models for a list of species, we use the function `getModels`. This is a composite of `as_fsvm`, `fsvm_train`, and `fsvm_train_lite`. Completed species models are stored along the provided 'file_path' parameter within the folders created by the 'group' and 'type' parameters and are saved as 'Genus species subspecies.RData'. The parameter 'lite' indicates whether to use `fsvm_train` or `fsvm_train_lite` to train the models. The parameter 'iterator' is used to restrict the list species to be modeled. "ALL" will use all unique species with in the provided field data, "Forage" will use only the species included in the data frame `fsvm::ForageSpecies`, and "MuleDeer", "Elk", "Moose", and "SageGrouse" will restrict the species list to only forage items corresponding to the named animal within the 'ForageSpecies' data frame. You can also specify your own customized subset of species to loop through

```
#Specify species list and where to save them
model_list <- c("Vaccinium scoparium","Solidago canadensis","Senecio triangularis","Rosa woodsii")
out_path <- "A:/Fine scale vegetation analysis/fsvm_package/Vignette_Examples"
```

```
#Train machine learning models
fsvm::getModels(fielddata = fielddata,
                file_path = out_path,
                iterator = model_list,
                group = "G2",
                type = "presence",
                lite = T,
                covs = covariates)
```

```
#Load models
savedMods <- dir(paste0(out_path,"/models/G2/presence"),full.names = T)
load(savedMods[1])
fsvm.res$BEST
> $max_ROC_caret
> [1] "XGBOOST"
>
> $min_ROCspread_caret
> [1] "NB_class"
>
> $max_AUC_test
> [1] "RF"
>
> $max_AUC_test_val
> [1] 0.7687001
```

Evaluate Trained Models

Now that you have completed model training, you can evaluate the SDMs using the `getSummary` function to examine details of the best models including appropriate statistics and run time for each species iteration.

```
#Summarize trained models
summary <- fsvm::getSummary(file_path=out_path,folder="G2",type="presence")
> [1] "iteration 1 of 4 complete"
> [1] "iteration 2 of 4 complete"
> [1] "iteration 3 of 4 complete"
> [1] "iteration 4 of 4 complete"
head(summary)
>                   Name EnoughData BestModel   TestAUC    ResSpec   ResSens
> 1         Rosa woodsii        Yes        RF 0.7687001 0.00000000 0.9983058
> 2 Senecio triangularis        Yes        RF 0.8969115 0.00000000 0.9996016
> 3  Solidago canadensis        Yes        RF 0.8956766 0.00000000 0.9996002
> 4  Vaccinium scoparium        Yes   XGBOOST 0.9636018 0.02234637 0.9729392
>   SampleSize user_time system_time elapsed_time
> 1        321    107.67        1.26        49.39
> 2        106     76.83        0.54        34.50
> 3        119     82.66        0.87        34.83
> 4        261      6.56        0.47         4.45
```

For details on getting predictions from trained 'fsvm' species distribution models, please see the vignette "FSVM Predictions".