Logistic Regression: Multinom and OVA/AVA

Import libraries as usual...

```
library(nnet)
```

Reading the data...

```
train_values <- read.csv(
  "../data/Richters_Predictor_Modeling_Earthquake_Damage_-_Train_Values.csv"
)
train_labels <- read.csv(
  "../data/Richters_Predictor_Modeling_Earthquake_Damage_-_Train_Labels.csv"
)
```

Multinom from NNet

We append the target variable to the full dataset, factor it and `relevel` it, a neccesary step for the `multinom` algorithm. Relevel simply makes one of the possible categories the "reference".

```
full_data = train_values
full_data$damage_grade = train_labels$damage_grade
full_data$damage_grade = factor(full_data$damage_grade, levels = c(1, 2, 3), labels=c("low damage", "me
full_data$damage_grade = relevel(full_data$damage_grade, ref=1)
```

Let us create a function that runs the `multinom algorithm`, and afterwards calculates the accuracy of the given model. Will take `n`, which is the number of rows we will use (set up at lower values when exploring so the algorithm does not take lots of time) and `formula`, in which we specify what variables we want to keep or discard.

```
prediction = function(n, formula){
  #build model
  model = multinom(formula, full_data[1:n,])

  #error calculation
  confMat = table(predict(model), full_data[1:n,]$damage_grade)


  accuracy = sum(diag(confMat))/sum(confMat)

  z = summary(model)$coefficients/summary(model)$standard.errors
  p <- (1 - pnorm(abs(z), 0, 1)) * 2
  print(p)

  return(accuracy)
}
```

This will make formula exploring much easier, since we can now just define n and formula, call the function and keep track of the differences in the model's accuracy.

Attaching the data so we can have access to the variables.

```
attach(full_data)
```

We'll start with 10 000 rows and all variables.

```
n = 10000 #260601
formula = damage_grade ~ .
accuracy1 = prediction(n, formula)
```

```
## # weights:  189 (124 variable)
## initial  value 10986.122887
## iter  10 value 9696.205911
## iter  20 value 8493.305728
## iter  30 value 8199.823452
## iter  40 value 7953.133742
## iter  50 value 7876.752849
## iter  60 value 7848.194210
## iter  70 value 7844.909446
## iter  80 value 7843.352040
## iter  90 value 7842.124532
## iter 100 value 7841.249515
## final  value 7841.249515
## stopped after 100 iterations
##                    (Intercept) building_id geo_level_1_id geo_level_2_id
## medium damage                0  0.08041031              0     0.02259624
## almost destructed            0  0.02491785              0     0.08873523
##                    geo_level_3_id count_floors_pre_eq age area_percentage
## medium damage          0.008189617                   0   0               0
## almost destructed      0.019761686                   0   0               0
##                    height_percentage land_surface_conditiono
## medium damage                      0                       0
## almost destructed                  0                       0
##                    land_surface_conditiont foundation_typei foundation_typer
## medium damage                            0                0                0
## almost destructed                        0                0                0
##                    foundation_typeu foundation_typew roof_typeq roof_typex
## medium damage                     0                0          0          0
## almost destructed                 0                0          0          0
##                    ground_floor_typem ground_floor_typev ground_floor_typex
## medium damage                       0                  0                  0
## almost destructed                   0                  0                  0
##                    ground_floor_typez other_floor_typeq other_floor_types
## medium damage                       0                 0                 0
## almost destructed                   0                 0                 0
##                    other_floor_typex positiono positions positiont
## medium damage                      0         0         0         0
## almost destructed                  0         0         0         0
##                    plan_configurationc plan_configurationd plan_configurationf
## medium damage                        0                   0                   0
## almost destructed                    0                   0                   0
```

```
##                   plan_configurationm plan_configurationn plan_configurationo
## medium damage                       0                   0                   0
## almost destructed                   0                   0                   0
##                   plan_configurationq plan_configurations plan_configurationu
## medium damage                       0                   0                   0
## almost destructed                   0                   0                   0
##                   has_superstructure_adobe_mud
## medium damage                                0
## almost destructed                            0
##                   has_superstructure_mud_mortar_stone
## medium damage                                       0
## almost destructed                                   0
##                   has_superstructure_stone_flag
## medium damage                                  0
## almost destructed                              0
##                   has_superstructure_cement_mortar_stone
## medium damage                                           0
## almost destructed                                       0
##                   has_superstructure_mud_mortar_brick
## medium damage                                       0
## almost destructed                                   0
##                   has_superstructure_cement_mortar_brick
## medium damage                                           0
## almost destructed                                       0
##                   has_superstructure_timber has_superstructure_bamboo
## medium damage                             0                         0
## almost destructed                         0                         0
##                   has_superstructure_rc_non_engineered
## medium damage                                       0
## almost destructed                                   0
##                   has_superstructure_rc_engineered has_superstructure_other
## medium damage                                   0                         0
## almost destructed                               0                         0
##                   legal_ownership_statusr legal_ownership_statusv
## medium damage                           0                       0
## almost destructed                       0                       0
##                   legal_ownership_statusw count_families has_secondary_use
## medium damage                           0              0                 0
## almost destructed                       0              0                 0
##                   has_secondary_use_agriculture has_secondary_use_hotel
## medium damage                                 0                       0
## almost destructed                             0                       0
##                   has_secondary_use_rental has_secondary_use_institution
## medium damage                             0                            0
## almost destructed                         0                            0
##                   has_secondary_use_school has_secondary_use_industry
## medium damage                             0                         0
## almost destructed                         0                         0
##                   has_secondary_use_health_post has_secondary_use_gov_office
## medium damage                                 0                          NaN
## almost destructed                             0                          NaN
##                   has_secondary_use_use_police has_secondary_use_other
## medium damage                              NaN                       0
## almost destructed                          NaN                       0
```

```
accuracy1
```

```
## [1] 0.5937
```

We got almost 0.6, which is not very bad, but could be a lot better.

Let's remove the geo levels 2 and 3, since they are too granular to give any information, and also the building id, which provides absolutely no information.

```
formula = damage_grade ~ . -building_id - geo_level_2_id - geo_level_3_id
accuracy2 = prediction(n, formula)
```

```
## # weights:  180 (118 variable)
## initial  value 10986.122887
## iter  10 value 9011.647781
## iter  20 value 8561.042569
## iter  30 value 8156.121618
## iter  40 value 7956.128547
## iter  50 value 7871.732758
## iter  60 value 7852.933739
## iter  70 value 7850.337896
## iter  80 value 7849.178637
## iter  90 value 7847.766319
## iter 100 value 7847.176647
## final  value 7847.176647
## stopped after 100 iterations
##                     (Intercept) geo_level_1_id count_floors_pre_eq        age
## medium damage     2.544614e-02   3.311509e-10        0.1324426269 0.08981588
## almost destructed 1.833441e-09   5.897890e-02        0.0002594504 0.15485961
##                   area_percentage height_percentage land_surface_conditiono
## medium damage           0.3067980         0.1899411               0.9959909
## almost destructed       0.8282815         0.2731493               0.5497962
##                   land_surface_conditiont foundation_typei foundation_typer
## medium damage                   0.2585174        0.4533627       0.05943847
## almost destructed               0.7985651        0.3755110       0.11005181
##                   foundation_typeu foundation_typew   roof_typeq roof_typex
## medium damage           0.04256024       0.34281780 5.713552e-03 0.23180421
## almost destructed       0.32882196       0.03074828 1.141854e-06 0.03471077
##                   ground_floor_typem ground_floor_typev ground_floor_typex
## medium damage              0.2659667       1.819557e-01          0.9676628
## almost destructed          0.8080716       4.414040e-06          0.9139848
##                   ground_floor_typez other_floor_typeq other_floor_types
## medium damage              0.7522545       0.002720146       0.005896677
## almost destructed          0.4745477       0.682173746       0.017892756
##                   other_floor_typex positiono positions   positiont
## medium damage            0.06341554 0.9164951 0.1144784 0.359668995
## almost destructed        0.64388351 0.8742566 0.4718908 0.002334726
##                   plan_configurationc plan_configurationd plan_configurationf
## medium damage              0.03282908        8.062414e-04                   0
## almost destructed          0.00000000        9.483059e-11                   0
##                   plan_configurationm plan_configurationn plan_configurationo
## medium damage               0.8888844                   0           0.3290001
## almost destructed           0.0000000                   0           0.1343017
```

```
##                   plan_configurationq plan_configurations plan_configurationu
## medium damage            1.856697e-03         0.050222304          0.18978461
## almost destructed        3.469891e-11         0.002388579          0.07088852
##                   has_superstructure_adobe_mud
## medium damage                    1.918790e-05
## almost destructed                2.137037e-05
##                   has_superstructure_mud_mortar_stone
## medium damage                            5.417888e-14
## almost destructed                        3.330669e-15
##                   has_superstructure_stone_flag
## medium damage                     2.134782e-03
## almost destructed                 4.447072e-06
##                   has_superstructure_cement_mortar_stone
## medium damage                                  0.44449383
## almost destructed                              0.04971995
##                   has_superstructure_mud_mortar_brick
## medium damage                            1.994248e-05
## almost destructed                        3.622453e-03
##                   has_superstructure_cement_mortar_brick
## medium damage                              4.915833e-01
## almost destructed                          6.967520e-08
##                   has_superstructure_timber has_superstructure_bamboo
## medium damage                 0.0609408316                0.41717652
## almost destructed             0.0000907402                0.03689253
##                   has_superstructure_rc_non_engineered
## medium damage                              0.091070740
## almost destructed                          0.006595444
##                   has_superstructure_rc_engineered has_superstructure_other
## medium damage                         0.008646976                0.8880423
## almost destructed                     0.099878468                0.9313274
##                   legal_ownership_statusr legal_ownership_statusv
## medium damage                   0.5559514               0.3449941
## almost destructed               0.3770801               0.5071857
##                   legal_ownership_statusw count_families has_secondary_use
## medium damage                   0.6545155   6.440687e-07                 0
## almost destructed               0.1996916   4.021506e-07                 0
##                   has_secondary_use_agriculture has_secondary_use_hotel
## medium damage                                 0                       0
## almost destructed                             0                       0
##                   has_secondary_use_rental has_secondary_use_institution
## medium damage                            0                  0.000000e+00
## almost destructed                        0                  6.619127e-09
##                   has_secondary_use_school has_secondary_use_industry
## medium damage                            0               1.634426e-11
## almost destructed                        0               2.533515e-05
##                   has_secondary_use_health_post has_secondary_use_gov_office
## medium damage                      7.435169e-08                             1
## almost destructed                  0.000000e+00                           NaN
##                   has_secondary_use_use_police has_secondary_use_other
## medium damage                                1               0.1547214
## almost destructed                          NaN               0.3546700
```

```
accuracy2
```

```
## [1] 0.5934
```

This provided almost no difference, so it was not a bad decision since we have less information to process but the same result. That tells us those variables were explaining nothing.

Let's keep subtracting variables and see what happens.

Based on the p values obtained in the model, `land_surface_condition` had a very high score. That means that the *confidence* that those variables are related with our target is 1 - p, so it must be really small. Removing that variable would result in a, hopefully better *accuracy*, but, at least, no change in it. We'll do that from now on, search for high p-value variables and remove them to see if we can *improve the model's accuracy*.

```
formula = damage_grade ~ . -building_id -
                           geo_level_2_id -
                           geo_level_3_id -
                           land_surface_condition

accuracy3 = prediction(n, formula)
```

```
## # weights:  174 (114 variable)
## initial  value 10986.122887
## iter  10 value 9007.371322
## iter  20 value 8532.088757
## iter  30 value 8117.029216
## iter  40 value 7935.789422
## iter  50 value 7867.790991
## iter  60 value 7854.673954
## iter  70 value 7852.282859
## iter  80 value 7851.094418
## iter  90 value 7849.723259
## iter 100 value 7849.163153
## final  value 7849.163153
## stopped after 100 iterations

## Warning in sqrt(diag(vc)): NaNs produced

## Warning in sqrt(diag(vc)): NaNs produced

##                     (Intercept) geo_level_1_id count_floors_pre_eq        age
## medium damage     2.003921e-02   3.125276e-10        0.1372687642 0.08230113
## almost destructed 5.527158e-09   6.043069e-02        0.0002423309 0.14862712
##                   area_percentage height_percentage foundation_typei
## medium damage           0.3015744         0.1815824        0.4572495
## almost destructed       0.8184114         0.2699842        0.3622289
##                   foundation_typer foundation_typeu foundation_typew
## medium damage           0.05722756       0.04005613       0.33393874
## almost destructed       0.10584389       0.31332759       0.03201083
##                     roof_typeq roof_typex ground_floor_typem ground_floor_typev
## medium damage     5.998269e-03 0.25379284          0.2480614       1.523478e-01
## almost destructed 1.166364e-06 0.03970416          0.7968017       2.959126e-06
##                   ground_floor_typex ground_floor_typez other_floor_typeq
## medium damage              0.9791581          0.7924857       0.002954413
## almost destructed          0.9206428          0.4874111       0.660545912
```

```
##                   other_floor_types other_floor_typex positiono positions
## medium damage           0.005739704        0.06056438 0.9160188 0.1294219
## almost destructed       0.015060514        0.65877974 0.8743857 0.4994394
##                     positiont plan_configurationc plan_configurationd
## medium damage     0.332884399          0.04030722        1.464766e-03
## almost destructed 0.001978831          0.00000000        2.375298e-09
##                   plan_configurationf plan_configurationm plan_configurationn
## medium damage                       0           0.7130182                   0
## almost destructed                   0           0.0000000                   0
##                   plan_configurationo plan_configurationq plan_configurations
## medium damage               0.4492057        3.092273e-03         0.049834024
## almost destructed           0.1726372        6.806300e-10         0.005376248
##                   plan_configurationu has_superstructure_adobe_mud
## medium damage               0.2521022                 1.696275e-05
## almost destructed           0.1539788                 2.070118e-05
##                   has_superstructure_mud_mortar_stone
## medium damage                            2.686740e-14
## almost destructed                        2.220446e-15
##                   has_superstructure_stone_flag
## medium damage                      2.226518e-03
## almost destructed                  4.794651e-06
##                   has_superstructure_cement_mortar_stone
## medium damage                                  0.41735368
## almost destructed                              0.04690691
##                   has_superstructure_mud_mortar_brick
## medium damage                            1.877741e-05
## almost destructed                        3.367862e-03
##                   has_superstructure_cement_mortar_brick
## medium damage                               5.032823e-01
## almost destructed                           8.083009e-08
##                   has_superstructure_timber has_superstructure_bamboo
## medium damage                  0.0689985941                0.41168733
## almost destructed              0.0001086419                0.03604595
##                   has_superstructure_rc_non_engineered
## medium damage                            0.102864822
## almost destructed                        0.008176718
##                   has_superstructure_rc_engineered has_superstructure_other
## medium damage                          0.009615322                0.8900296
## almost destructed                      0.114334440                0.9183389
##                   legal_ownership_statusr legal_ownership_statusv
## medium damage                   0.5875776               0.3088739
## almost destructed               0.3690519               0.4835536
##                   legal_ownership_statusw count_families has_secondary_use
## medium damage                   0.6952685   5.806736e-07                 0
## almost destructed               0.2111885   3.557782e-07                 0
##                   has_secondary_use_agriculture has_secondary_use_hotel
## medium damage                                 0                       0
## almost destructed                             0                       0
##                   has_secondary_use_rental has_secondary_use_institution
## medium damage                            0                  0.000000e+00
## almost destructed                        0                  5.210237e-10
##                   has_secondary_use_school has_secondary_use_industry
## medium damage                            0               3.708145e-14
## almost destructed                        0               1.359365e-06
```

```
##                   has_secondary_use_health_post has_secondary_use_gov_office
## medium damage                      4.430815e-09                           NaN
## almost destructed                  0.000000e+00                           NaN
##                   has_secondary_use_use_police has_secondary_use_other
## medium damage                              NaN                0.1277590
## almost destructed                          NaN                0.3025609
```

```
accuracy3
```

```
## [1] 0.5927
```

For the sake of common sense, political variables like `legal_ownership_status` o the `has_secondary_use_*` family would probably not add much to the concept of a building being able to deal with an earthquake, so lets remove them. Some of these variables are also categorical, so we are removing much more than it seems, since the model tries with every possible combination.

```
formula = damage_grade ~ . -building_id -
                           geo_level_2_id -
                           geo_level_3_id -
                           land_surface_condition -
                           legal_ownership_status -
                           has_secondary_use_institution -
                           has_secondary_use_health_post -
                           has_secondary_use_use_police -
                           has_secondary_use_other

accuracy4 = prediction(n, formula)
```

```
## # weights:  153 (100 variable)
## initial  value 10986.122887
## iter  10 value 9013.990341
## iter  20 value 8623.682045
## iter  30 value 8149.666955
## iter  40 value 7946.290462
## iter  50 value 7874.001112
## iter  60 value 7864.926849
## iter  70 value 7862.781223
## iter  80 value 7862.440690
## iter  90 value 7862.328688
## iter 100 value 7862.238453
## final  value 7862.238453
## stopped after 100 iterations
```

```
## Warning in sqrt(diag(vc)): NaNs produced
```

```
## Warning in sqrt(diag(vc)): NaNs produced
```

```
##                    (Intercept) geo_level_1_id count_floors_pre_eq        age
## medium damage     8.649062e-03   4.577398e-10        0.1405732562 0.07938902
## almost destructed 1.948928e-10   6.696710e-02        0.0002685905 0.13745134
##                   area_percentage height_percentage foundation_typei
## medium damage           0.3496349         0.2057425        0.4720371
```

```
## almost destructed       0.7226673         0.3062347        0.3553855
##                    foundation_typer foundation_typeu foundation_typew
## medium damage           0.05867776       0.04345532       0.33021204
## almost destructed       0.10862484       0.33190822       0.03189147
##                      roof_typeq roof_typex ground_floor_typem ground_floor_typev
## medium damage     6.724986e-03  0.2168686          0.2331919       1.434980e-01
## almost destructed 1.388688e-06  0.0310060          0.7825874       2.238632e-06
##                    ground_floor_typex ground_floor_typez other_floor_typeq
## medium damage              0.9988191          0.7996401       0.002800905
## almost destructed          0.8860901          0.4851607       0.641646981
##                    other_floor_types other_floor_typex positiono positions
## medium damage           0.005476757        0.06224622 0.8810632 0.1302970
## almost destructed       0.014006880        0.68796159 0.8554294 0.4651736
##                      positiont plan_configurationc plan_configurationd
## medium damage     0.327530537          0.03921285        1.384365e-03
## almost destructed 0.002201971          0.00000000        7.864327e-10
##                    plan_configurationf plan_configurationm plan_configurationn
## medium damage                        0           0.6583978                   0
## almost destructed                    0           0.0000000                   0
##                    plan_configurationo plan_configurationq plan_configurations
## medium damage               0.4142918        3.881796e-03         0.045444534
## almost destructed           0.1424293        4.199523e-10         0.004572644
##                    plan_configurationu has_superstructure_adobe_mud
## medium damage                0.2551985                 1.889195e-05
## almost destructed            0.1254244                 1.854579e-05
##                    has_superstructure_mud_mortar_stone
## medium damage                            5.351275e-14
## almost destructed                        5.551115e-15
##                    has_superstructure_stone_flag
## medium damage                        2.24297e-03
## almost destructed                    5.44452e-06
##                    has_superstructure_cement_mortar_stone
## medium damage                                 0.41555841
## almost destructed                             0.04432849
##                    has_superstructure_mud_mortar_brick
## medium damage                           0.0000157563
## almost destructed                       0.0030236047
##                    has_superstructure_cement_mortar_brick
## medium damage                              5.605388e-01
## almost destructed                          1.037987e-07
##                    has_superstructure_timber has_superstructure_bamboo
## medium damage                 0.0819342704                0.37903008
## almost destructed             0.0001192877                0.03316229
##                    has_superstructure_rc_non_engineered
## medium damage                              0.094393618
## almost destructed                          0.006766134
##                    has_superstructure_rc_engineered has_superstructure_other
## medium damage                         0.008993038                0.8986641
## almost destructed                     0.118494916                0.9228185
##                    count_families has_secondary_use
## medium damage        7.604958e-07        0.04494543
## almost destructed    2.557502e-07        0.76417528
##                    has_secondary_use_agriculture has_secondary_use_hotel
## medium damage                         0.8985309              0.06310092
```

```
## almost destructed                    0.1421835             0.81878053
##                   has_secondary_use_rental has_secondary_use_school
## medium damage                0.0006781564                        0
## almost destructed            0.0669564762                        0
##                   has_secondary_use_industry has_secondary_use_gov_office
## medium damage                      0.9058610                          NaN
## almost destructed                  0.4620789                          NaN
```

```
accuracy4
```

```
## [1] 0.5913
```

OVA and AVA methods

```
library(regtools)
```

```
## Loading required package: FNN

## Loading required package: mvtnorm

## Loading required package: dummies

## dummies-1.5.6 provided by Decision Patterns

## Loading required package: sandwich

##
##
##
##
##
## **********************
##
##
##
## Latest version of regtools at GitHub.com/matloff
##
##
## Type "?regtools" for function list.
```

In order for the data to work with the regtools library, we must convert all columns to numeric or int variables. So, we factor the target variable and then parse it as numeric.

```
# In order for the data to work with the regtools library, we must convert all columns
# to numeric or int variables. So, we factor the target variable and then parse it as numeric.
full_data = train_values
full_data$damage_grade = train_labels$damage_grade
full_data$damage_grade = factor(full_data$damage_grade, levels = c(1, 2, 3), labels=c("low damage", "me
full_data$damage_grade = as.numeric(full_data$damage_grade) - 1
```

We do the same with all non-numeric variables

```
full_data[sapply(full_data, is.character)] = lapply(full_data[sapply(full_data, is.character)], as.facto
full_data[sapply(full_data, is.factor)] = lapply(full_data[sapply(full_data, is.factor)], function(x) {a
```

Let's start with the classifiers.

One Vs All

For it to work, we must set the target column as the last one in the dataframe, so we can just make a subset of the dataset in the order we need. Since the target was already the last one, its index is 40, but when making that selection it is necessary.

```
ovatrn = ovalogtrn(3, full_data[ ,c(2,5:39, 40)])
```

Finally, to predict, we get rid of the target column and use the predict helper function.

```
ovaypred <- ovalogpred(ovatrn, full_data[,c(2,5:39)])
```

The mean function over a boolean vector gives us the proportion of true/all values in the vector. Essentially: the accuracy.

```
mean(ovaypred == full_data$damage_grade)
```

```
## [1] 0.581832
```

Quadratic data

Let us try with quadratic data, which may exaggerate some of the features in the variables and, possibly, make it a bit easier for the algorithm.

Take all the columns we are interested in and square them. Save that subset.

```
quadratic_data = full_data[,c(2,5:39)]^2
```

Now the subset lacks the target variable, so we append it before it is passed to the function via cbind (column bind).

```
qovadata = ovalogtrn(3, cbind(quadratic_data, full_data$damage_grade))
```

We predict as usual and get our boolean vector.

```
qovaypred <- ovalogpred(qovadata, quadratic_data)
```

```
mean(qovaypred == full_data$damage_grade)
```

```
## [1] 0.5793992
```

All vs All

For it to work, we must pass in a matrix, not a dataframe. Let us convert the data into matrix form, again, with the subset we are interested in, and the target value last.

```
data_matrix = data.matrix(full_data[ ,c(2,5:39, 40)])
```

Call ava train function and classify.

```
avatrn = avalogtrn(3, data_matrix)
```

Predict as usual, but keeping the target variable out. No need for matrix form in this function.

```
avaypred <- avalogpred(3, avatrn, data.matrix(full_data[,c(2,5:39)]))
```

```
mean(avaypred == full_data$damage_grade)
```

```
## [1] 0.5822656
```

Quadratic Data

Let us try again with quadratic data. The quadratic version was defined before, so we can just use it.

```
data_matrix = data.matrix(cbind(quadratic_data, full_data$damage_grade))
```

```
avatrn = avalogtrn(3, data_matrix)
avaypred <- avalogpred(3, avatrn, data.matrix(full_data[,c(2,5:39)]))
mean(avaypred == full_data$damage_grade)
```

```
## [1] 0.575017
```

It seems like the quadratic strategy did not help much. This method did not help much either, since the results against `multinom` are almost exactly the same.