```r
library(RSSL)
library(magrittr)
library(ggplot2)

# Plotting 2D classifiers
data_2gauss <- generate2ClassGaussian(n=500,d=2,var=0.2,expected=FALSE) %>%
  add_missinglabels_mar(formula=Class~.,prob=0.98)
problem <- data_2gauss %>% df_to_matrices(Class~.)

g_emlda <- EMLinearDiscriminantClassifier(problem$X,problem$y,problem$X_u)
ggplot(data_2gauss,aes(x=X1,y=X2,shape=Class,color=Class)) +
  geom_point() +
  geom_classifier("EMLDA"=g_emlda)

# Generate Learning Curve
datasets <- list("2 Gaussian Expected" =
                   generate2ClassGaussian(n=1000,d=2,expected=TRUE),
                 "2 Gaussian Non-Expected" =
                   generate2ClassGaussian(n=1000,d=2,expected=FALSE))
formulae <- list("2 Gaussian Expected" = formula(Class~.),
                 "2 Gaussian Non-Expected" = formula(Class~.))

classifiers <- list("LS" = function(X,y,X_u,y_u) {
                      LeastSquaresClassifier(X,y)},
                    "ICLS" = function(X,y,X_u,y_u) {
                      ICLeastSquaresClassifier(X,y,X_u)},
                    "EMLS" = function(X,y,X_u,y_u) {
                      EMLeastSquaresClassifier(X,y,X_u)},
                    "SLLS" = function(X,y,X_u,y_u) {
                      SelfLearning(X,y,X_u,
                                   method = LeastSquaresClassifier)})

measures = list("Error" = measure_error,
                "Loss test" = measure_losstest)

curve <- LearningCurveSSL(formulae, datasets, classifiers, measures,
                          type ="unlabeled", mc.cores=1,
                          n_l=10,sizes = 2^(0:10),repeats=200)
plot(curve)
```