**REPORT: BAYESIAN AND SVM**

**PART 1: BAYESIAN  
  
Implement the Breast Cancer Exercise in the presentation in GeNIe or Hugin and also compute the posterior probability manually.**

**Women have breast cancer:** P(Cancer) = 0.01

|  |  |  |
| --- | --- | --- |
| **ZZ** | **CANCER** | **NOT CANCER** |
| **TEST** | P(Test|Cancer)= 0.8 | P(Test|not Cancer)=0.096 |
| **NOT TEST** | P(not Test|Cancer)=0.2 | P(not Test|not Cancer)=0.904 |

**Bayes Computation:**

P(Cancer|Test) = P(Test|Cancer) \* P(Cancer)/ P(Test)

P(Test) = P(Test and Cancer) + P(Test and not Cancer) ( Joint Condition)

= P(Cancer) \* P(Test|Cancer) + P(not Cancer) \* P(Test|not Cancer)

= 0.01\*0.8 + 0.99\*0.096

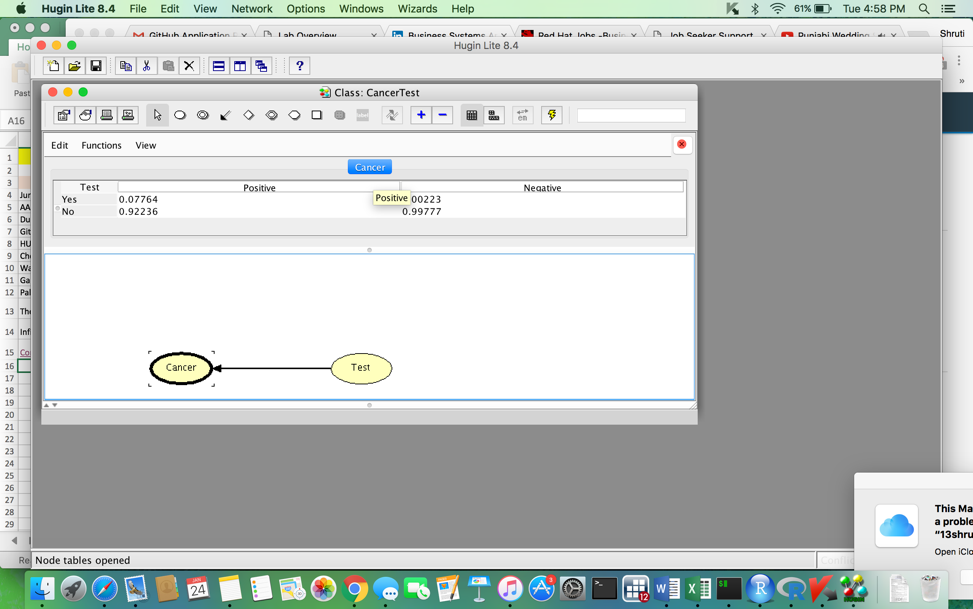
= 0.008 + 0.095

= 0.10304

P(Cancer|Test) = (P(Test|Cancer) \* P(Cancer)) / P(Test)

= (0.8\*0.01)/0.10304

= 0.0786  
  
Approximately **8%** chances of having Breast Cancer, if the result is positive.



**Compute also p(D | S1, S2) manually and in GeNIe using your own data for p(D), p(D | S1), and so on.  
  
Probability of having a disease (D) based on Symptom 1 (S1) and Symptom 2 (S2):**P(D) = 0.30

P(Not D) = 0.70

P(S1 | D) = 0.6

P(Not S1 | D) = 0.4

P(S1 | Not D) = 0.05

P(Not S1 | Not D) = 0.95

P(S2 | D) = 0.23

P(Not S2 | D) = 0.77

P(S2 | Not D) = 0.01

P(Not S2 | Not D) = 0.99

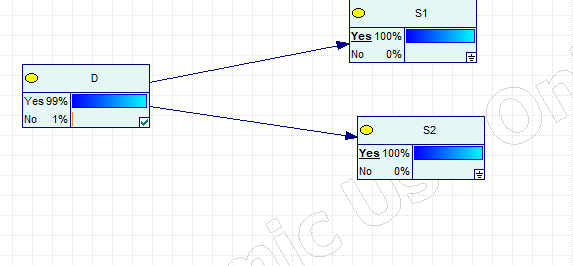
**Bayes Computation:**  
  
P(D | S1&S2) = [P(S2|D)\*P(S1|D)\*P(D)] / [P(S2|D)\*P(S1|D)\*P(D)] + [P(not D)\*P(S1|Not D)\*P(S2|Not D)]

= [0.23\*0.6\*0.30] / [0.23\*0.6\*0.30] + [0.70\*0.05\*0.01]

= 0.0414/(0.0414+0.00035)

= 0.9916

Approximately **99%** chances of having disease, if both symptoms are present.



**PART 2: SUPPORT VECTOR MACHINES (SVM)  
  
For this small assignment, the code is already there (assignment\_SVM.R), you just need to explain what the code does. At the end try to find the best parameters for the SVM function. ALSO, run the same example in Orange. Run all the other attached SVM examples.**

#Load caret and e1071 packages

#Assign the GermanCredit dataset to a variable called dataset

library(caret) #holds functions to build predictive model

library(e1071) #to use SVM in R

data(GermanCredit)

dataset = GermanCredit

#(ADDED)

#Print the summary and content of dataset for better understanding

summary(dataset)

dataset

#View the structure of the GermanCredit dataset

#'as.dataframe' checks whether the object is a dataframe and lapply returns a list of the elements that corresponds to the given range of values and scales the column

#The outcome would give the list of all the rows and columns between 1 to 7 in the dataset

#Display the structure to see the output

str(dataset)

dataset[,1:7] = as.data.frame(lapply(dataset[,1:7], scale))

str(dataset)

#'sample\_index' will display random 200 values out of the 1000

#Create test and train datasets with the selected data and print

sample\_index = sample(1000, 200)

test\_dateset = dataset[sample\_index,]

test\_dateset

train\_dateset = dataset[-sample\_index,]

train\_dateset

#Create SVM to get values from the RADIAL kernel and show summary

radialmodel <- svm(Class~., kernel = "radial", data = train\_dateset, ranges = list(gamma=2^(-15:3) , cost=2^(-5:15)), scale = F)

summary(radialmodel)

#Create SVM to get values from the LINEAR kernel and show summary

linearmodel <- svm(Class~., kernel = "linear", data = train\_dateset, ranges = list(gamma=2^(-15:3), cost=2^(-5:15)), scale = F)

summary(linearmodel)

#Create SVM to get values from the POLYNOMIAL kernel and show summary

polymodel <- svm(Class~.,kernel = "polynomial", data = train\_dateset, ranges = list(gamma=2^(-15:3), cost=2^(-5:15)), scale = F)

summary(polymodel)

#Create SVM to get values from the SIGMOID kernel and show summary

sigmoidmodel <- svm(Class~.,kernel = "sigmoid", data = train\_dateset, ranges = list(gamma=2^(-15:3), cost=2^(-5:15)), scale = F)

summary(sigmoidmodel)

#We will use a generic function called 'Predict' for predictions from the results of various model fitting functions.

#The function invokes particular methods which depend on the class of the first argument.

predictions1 <- predict(radialmodel, test\_dateset[-10])

predictions1

predictions2 <- predict(linearmodel, test\_dateset[-10])

predictions2

predictions3 <- predict(polymodel, test\_dateset[-10])

predictions3

predictions4 <- predict(sigmoidmodel, test\_dateset[-10])

#'Table' function creates and displays tabular results of categorical variables from the test dataset

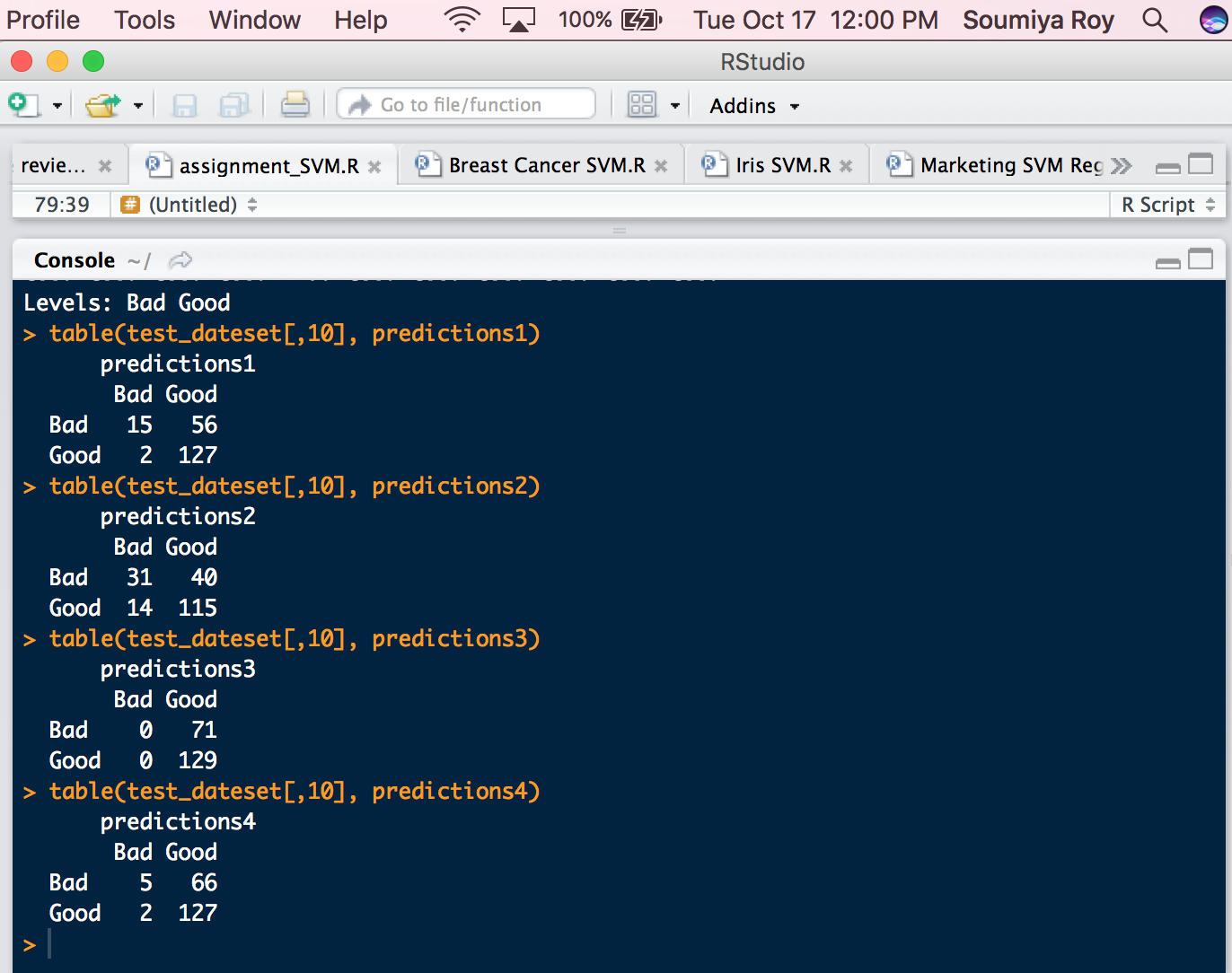
table(test\_dateset[,10], predictions1)

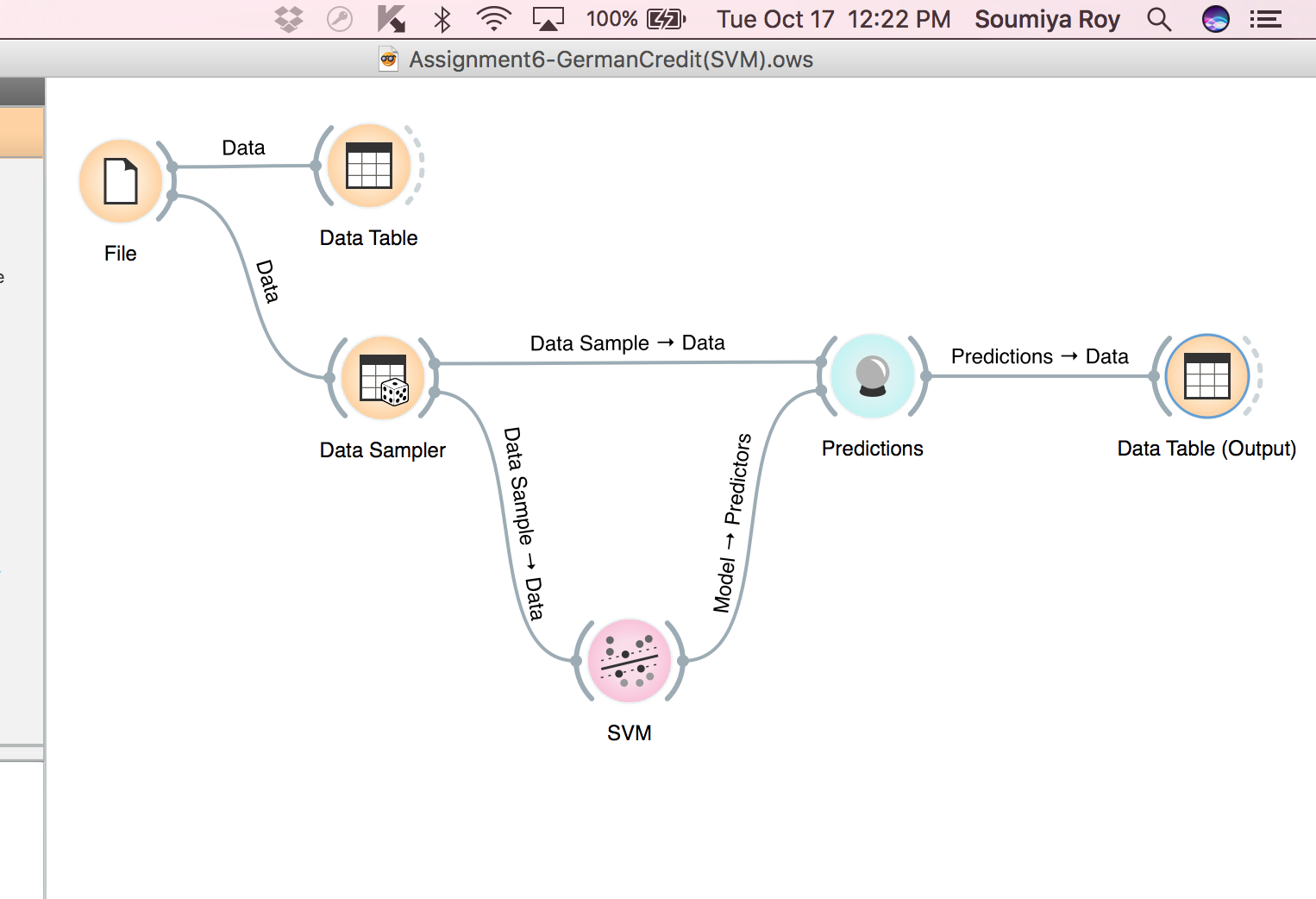
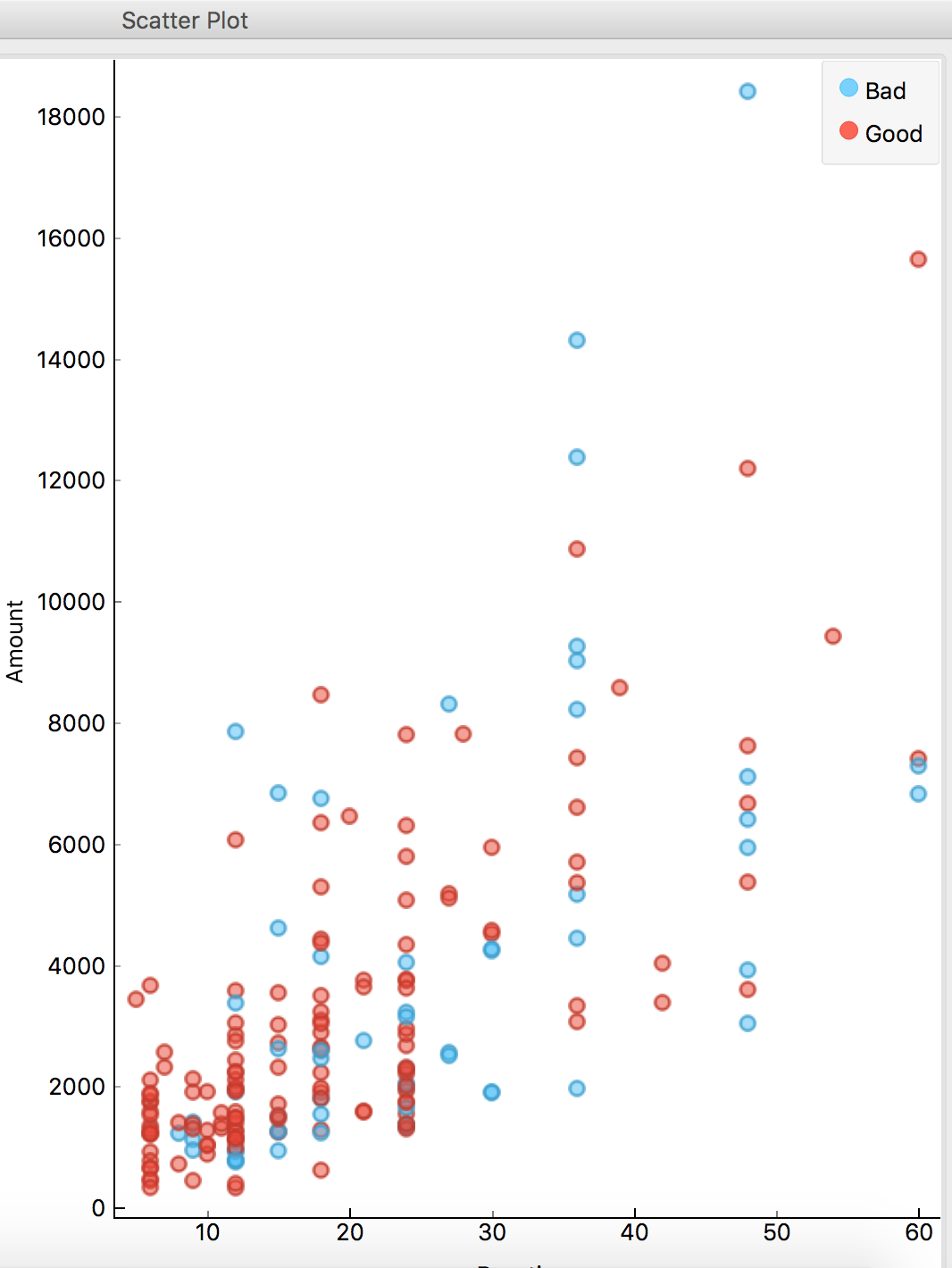
table(test\_dateset[,10], predictions2)

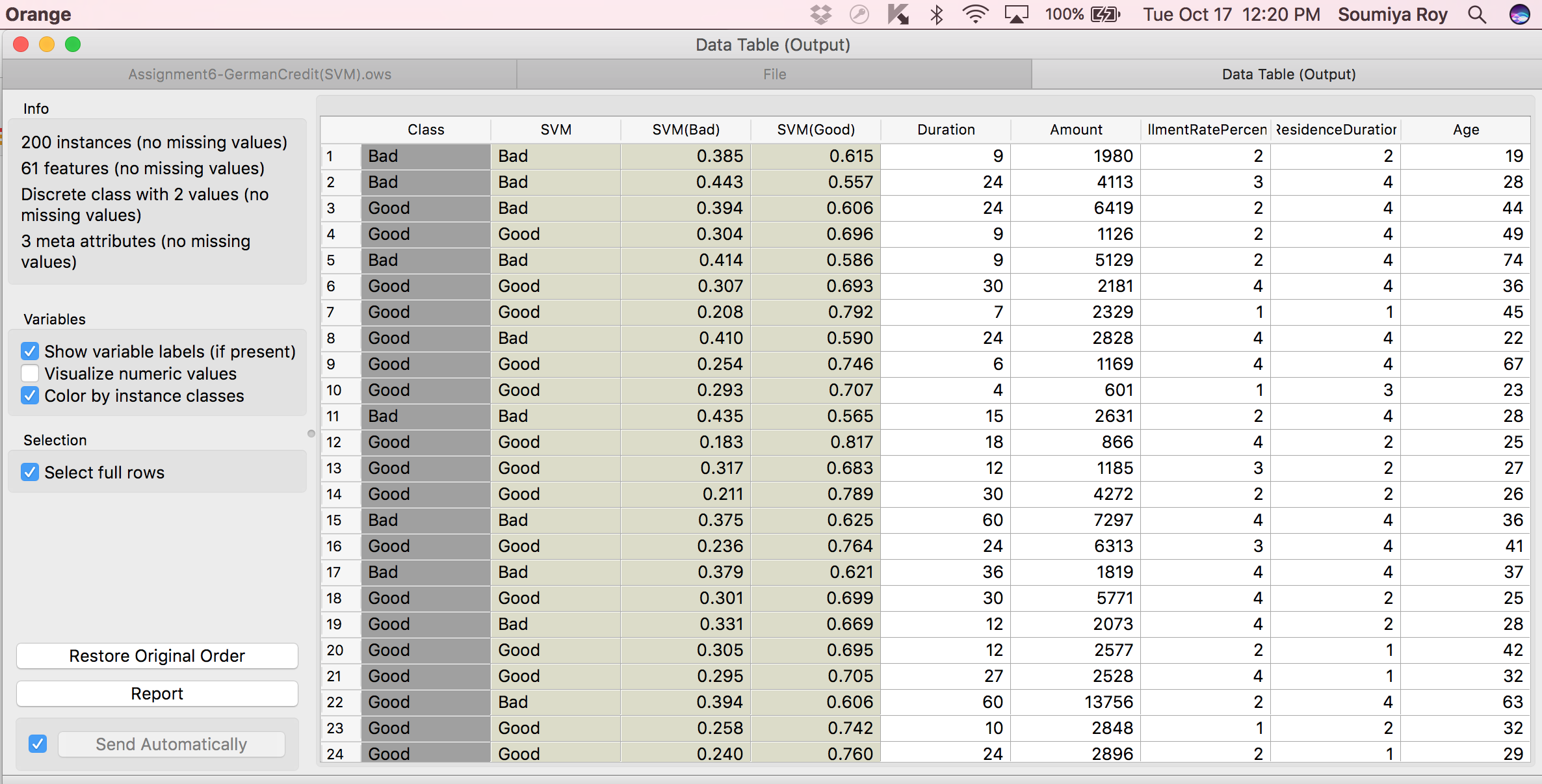
table(test\_dateset[,10], predictions3)

table(test\_dateset[,10], predictions4)

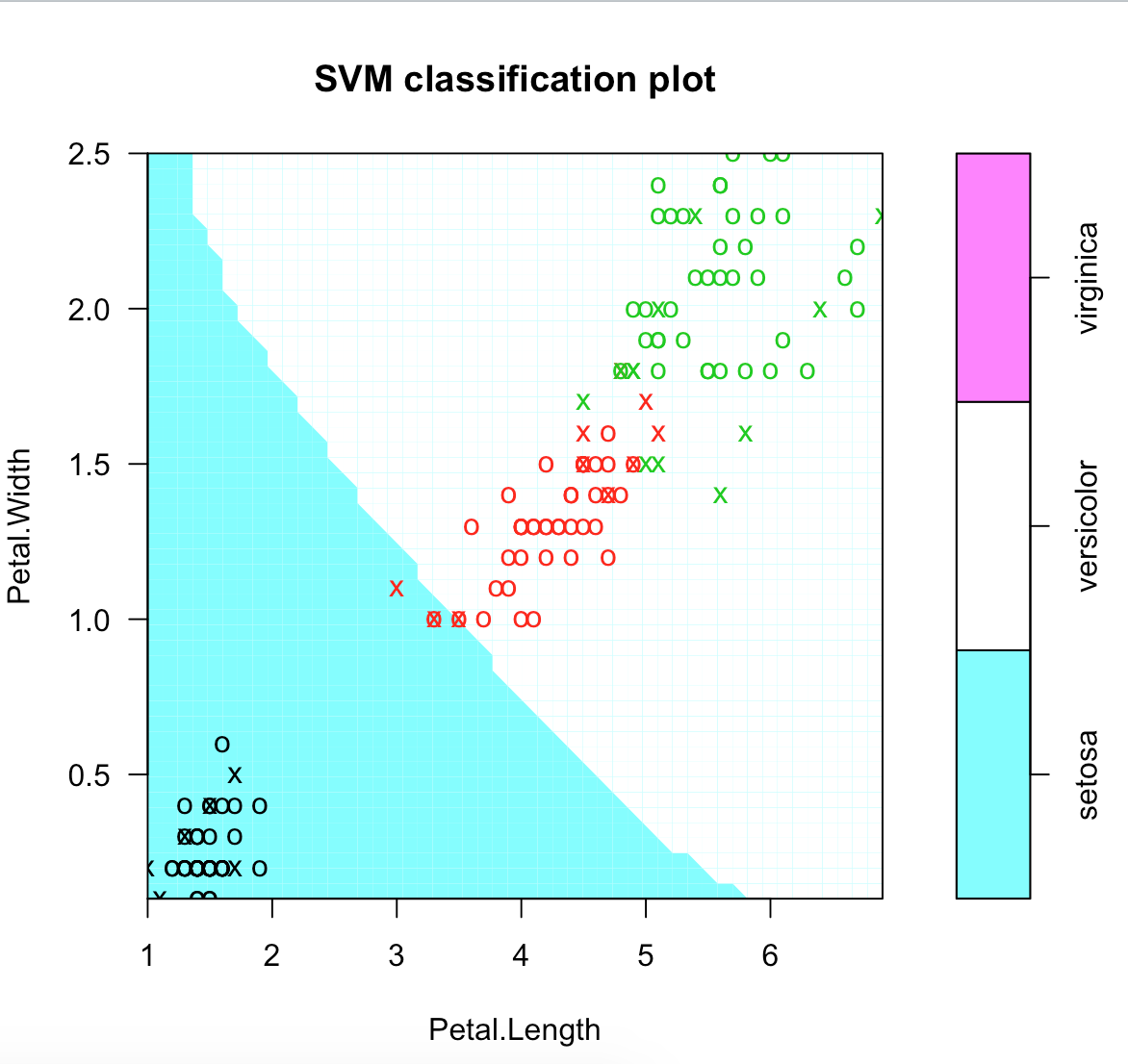
**Confusion Matrix of the SVM models:**

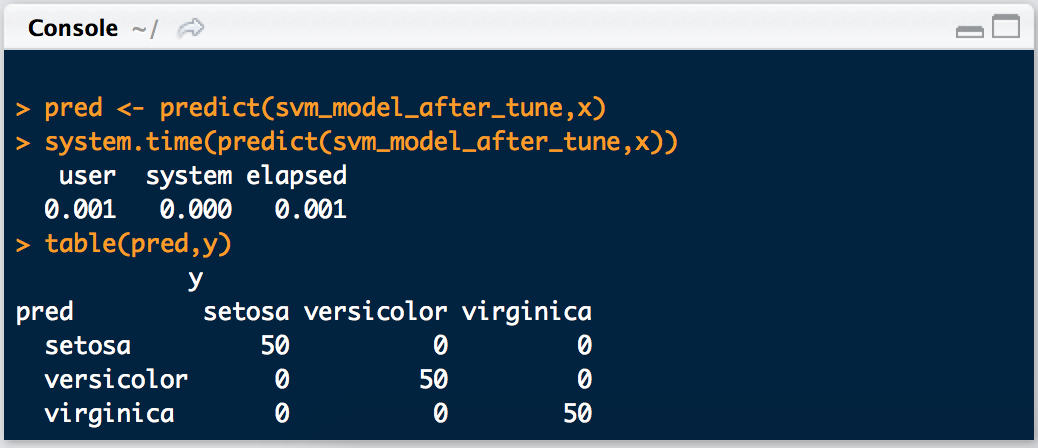
****

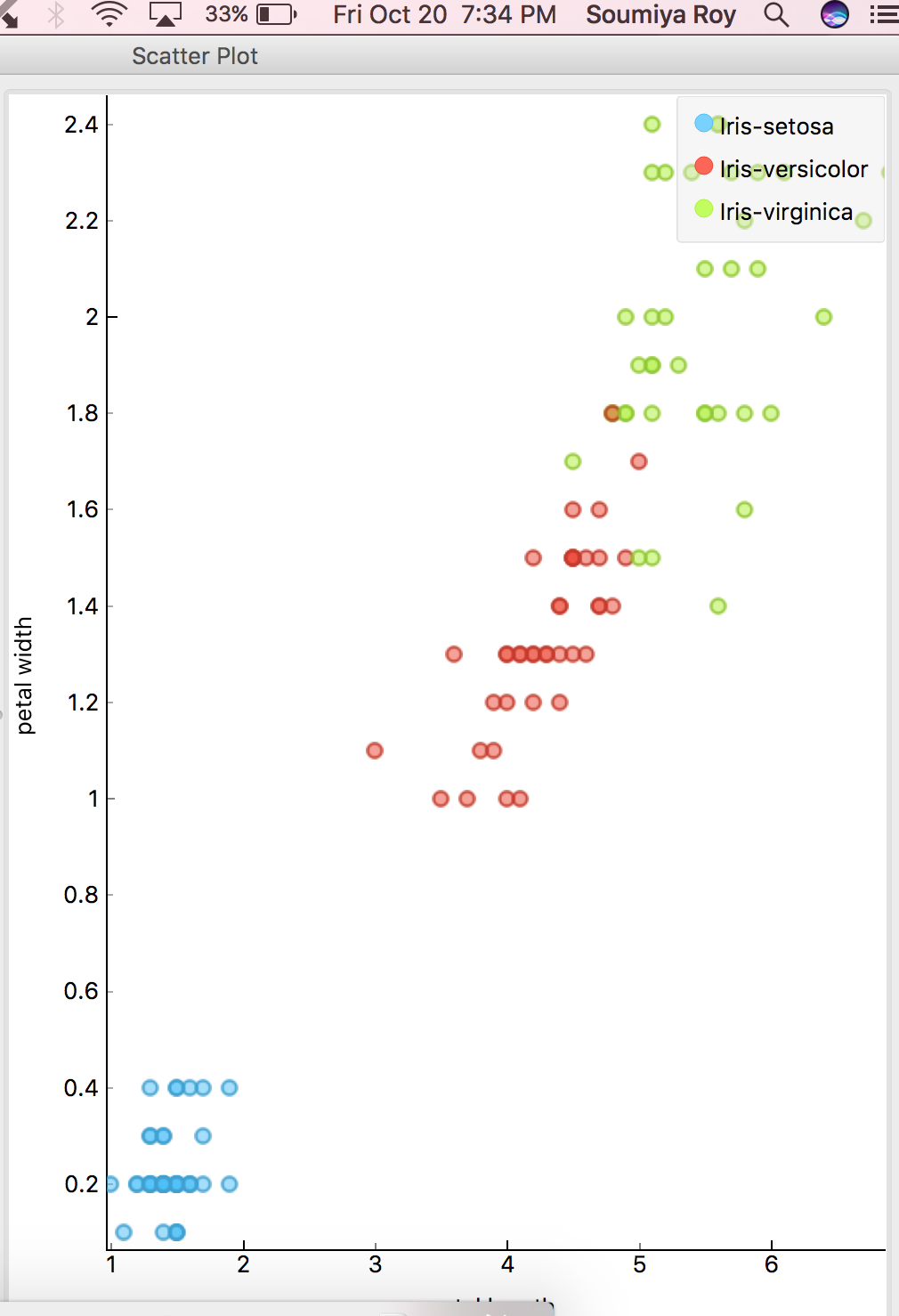
****Orange:

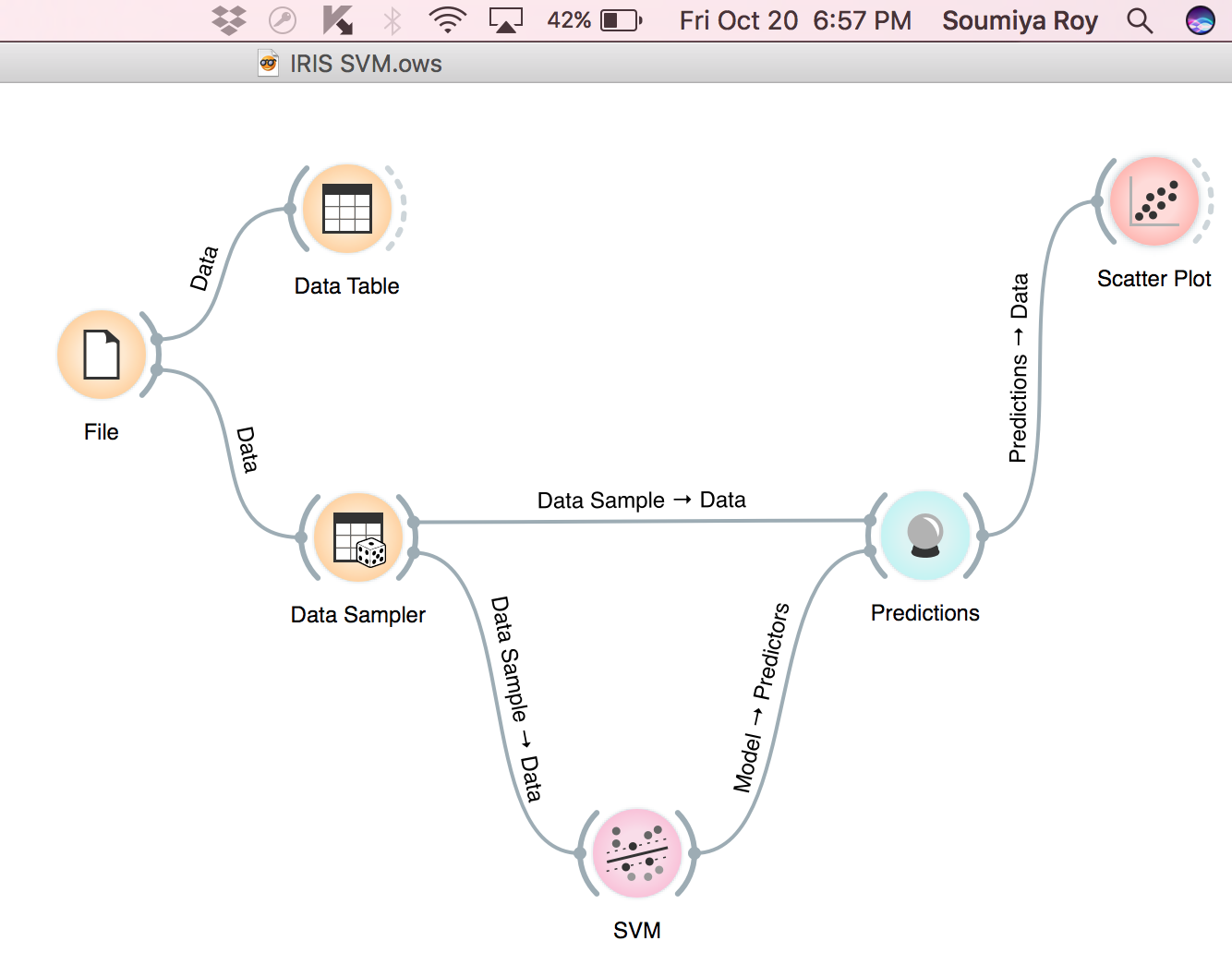
****

**IRIS SVM:**

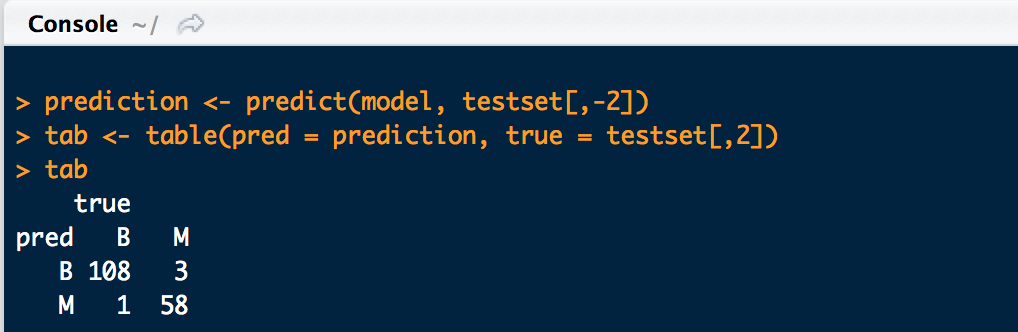
****

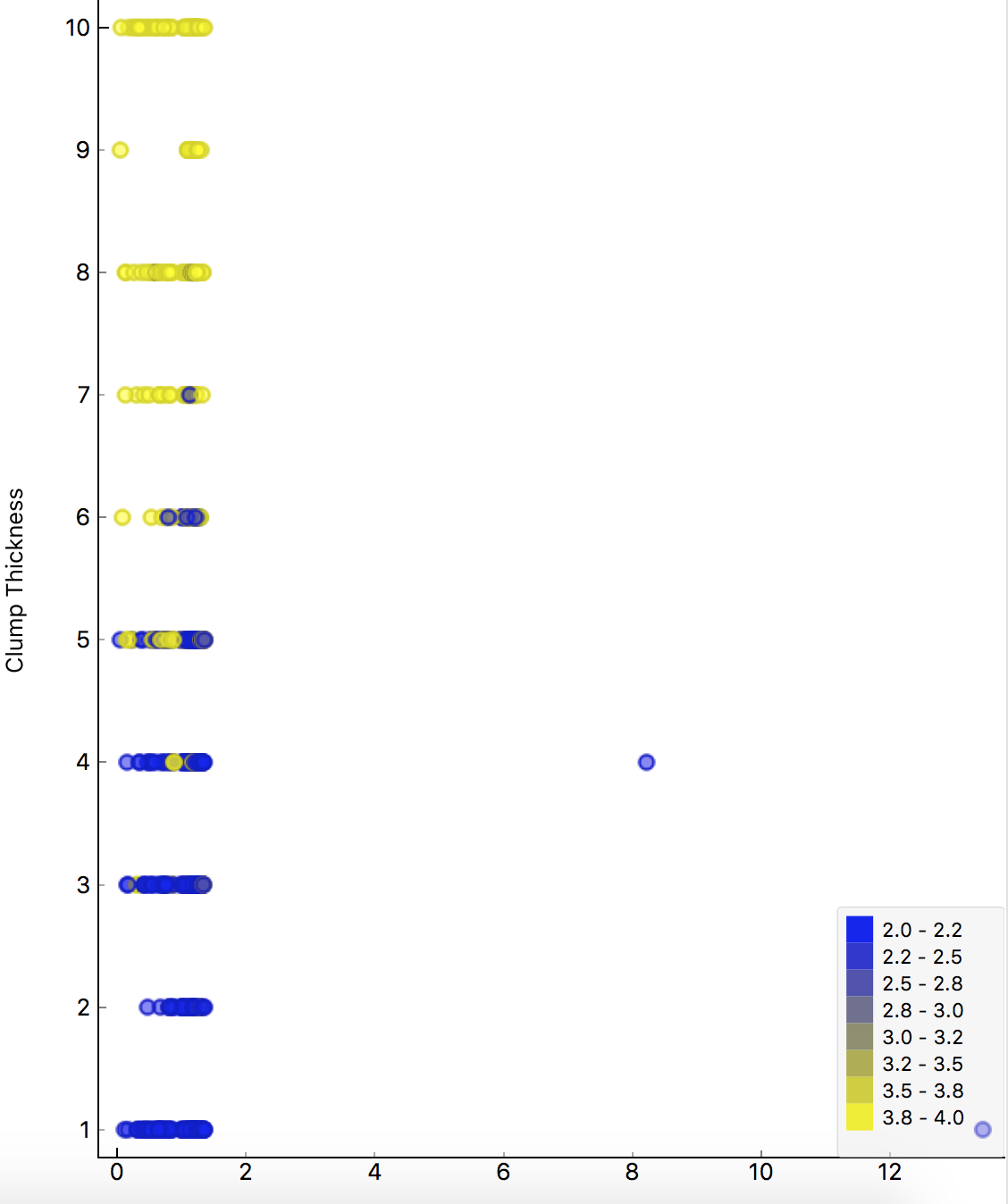
****

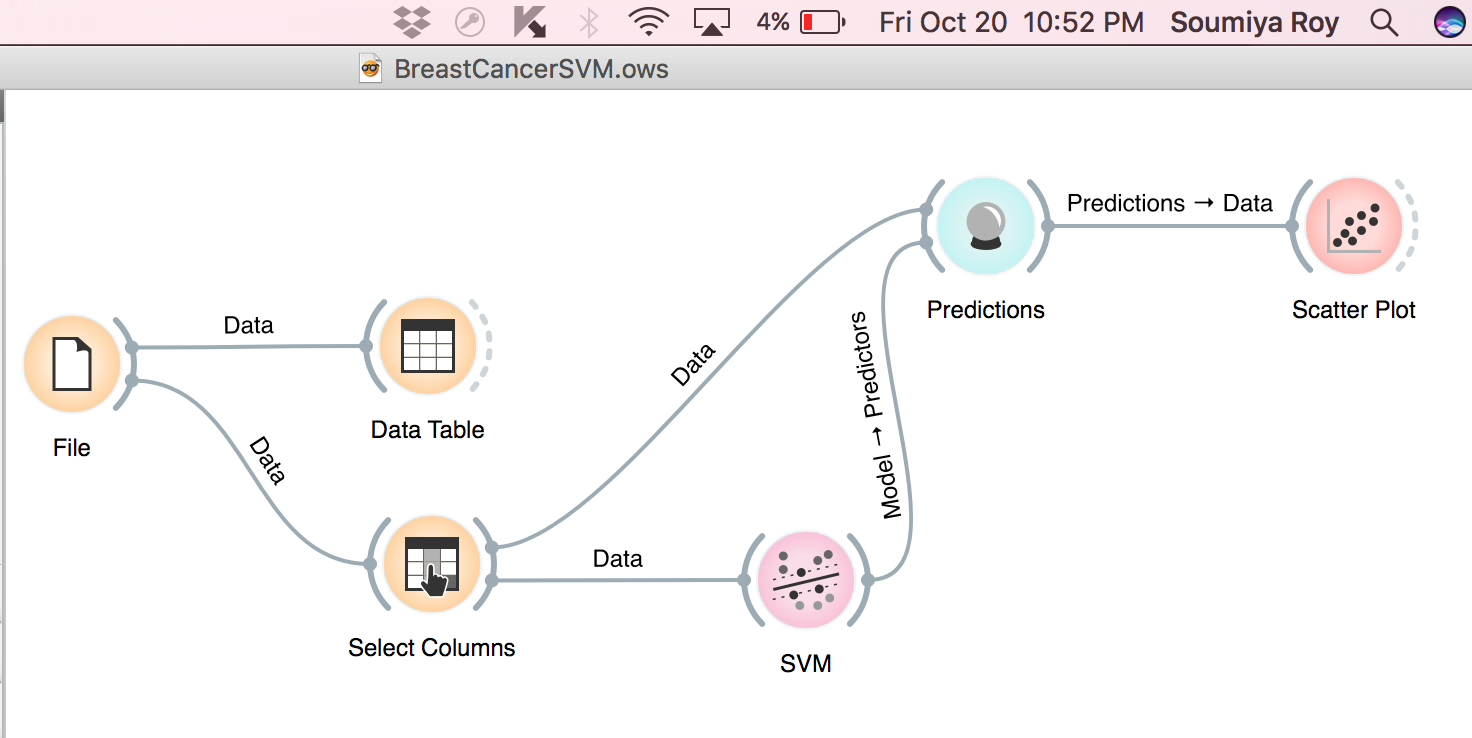
****

****

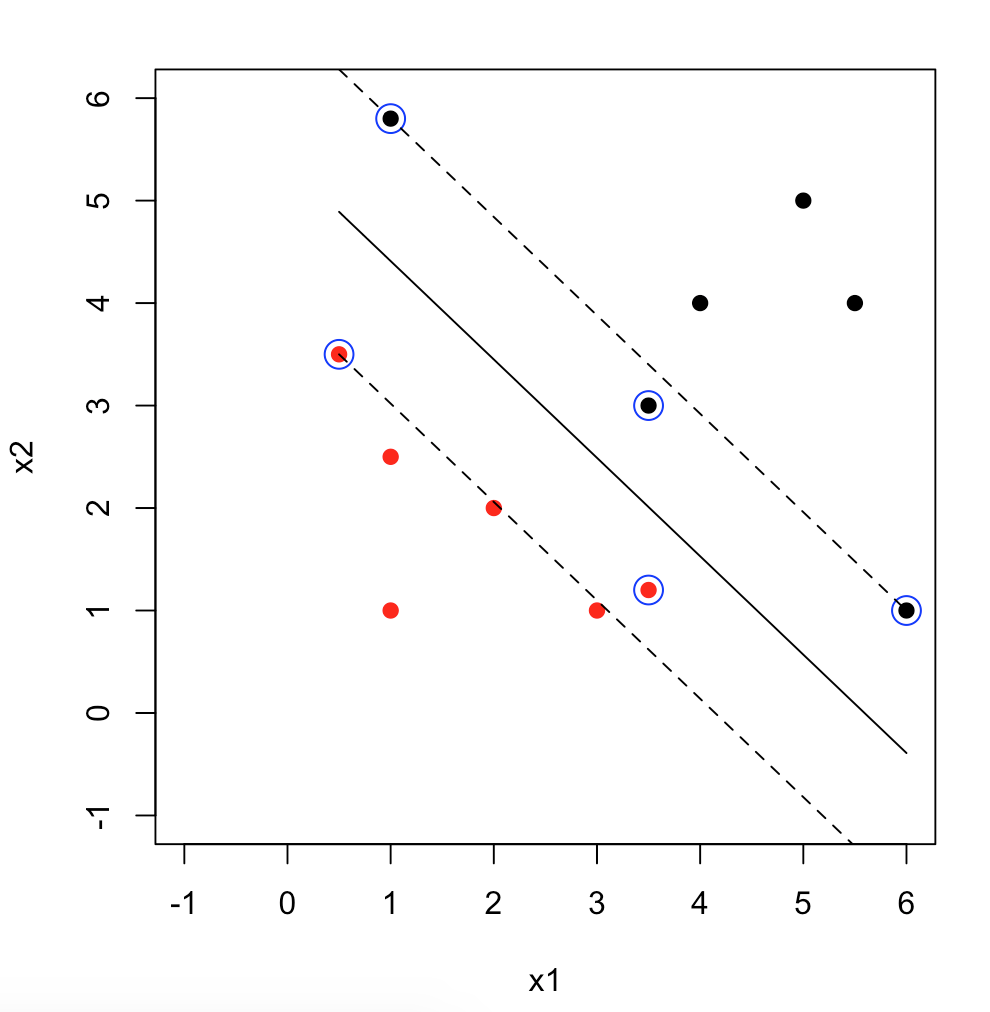
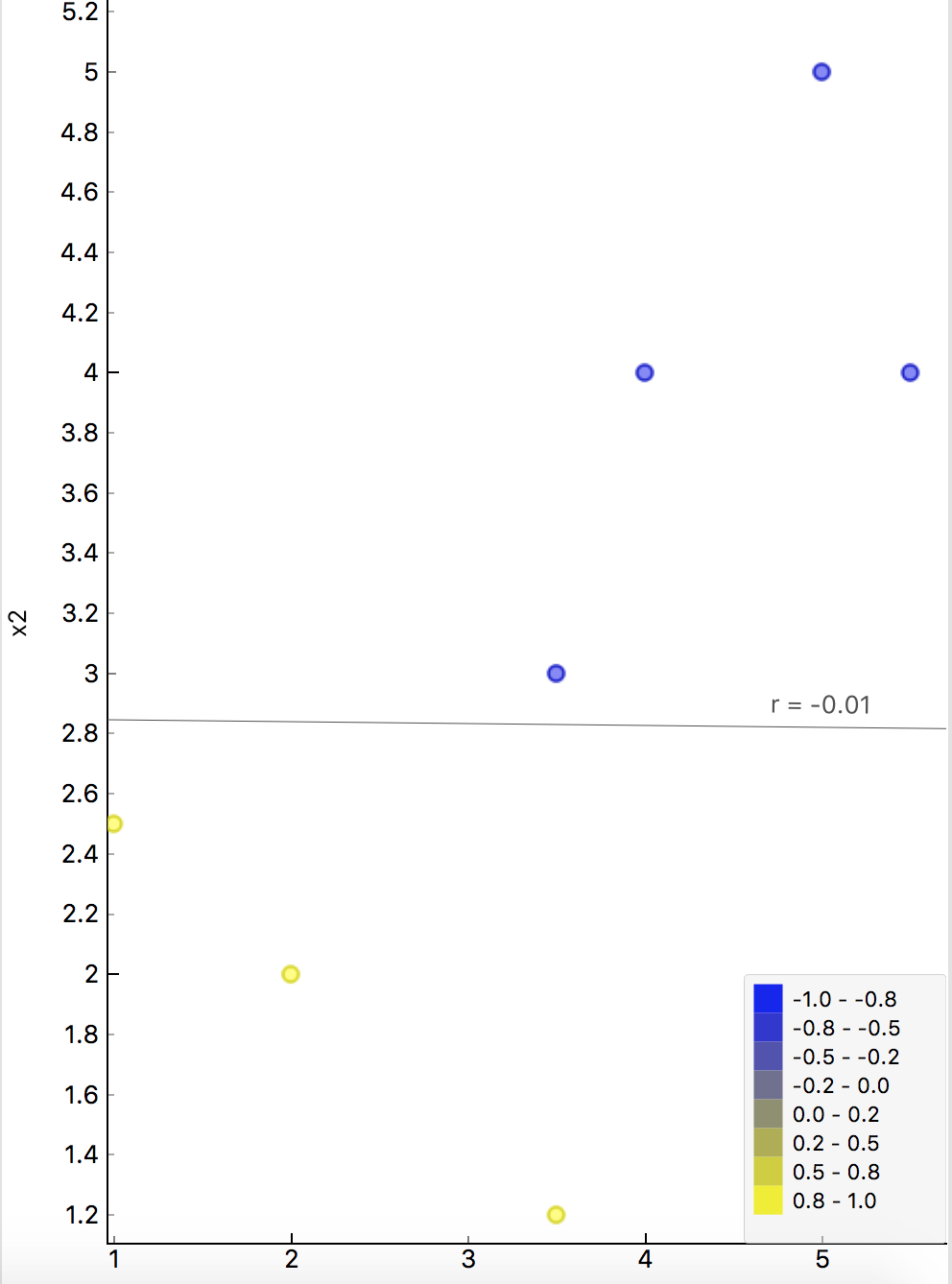
**Breast Cancer SVM:**

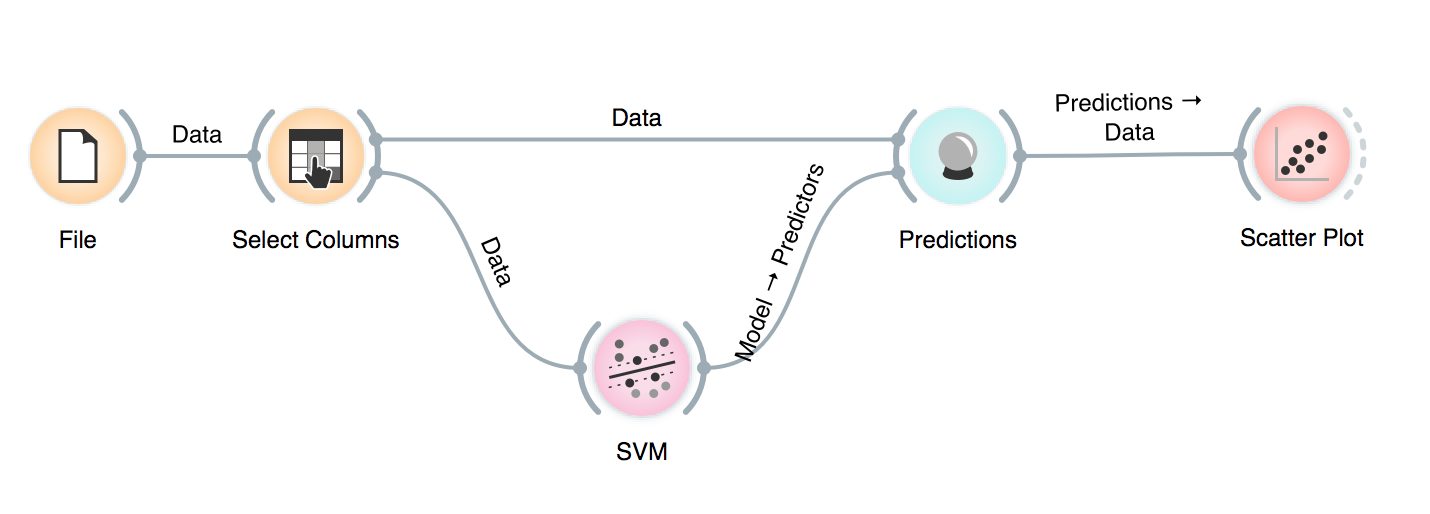
****

****

****

**Number of Support Vectors:**

****

****