**Big Data Spring 2018**

**Project 3**

Nicolas Eldering

Yifan Chen

**Member 1: Nicolas Eldering**  
My teammate and I agree that I handled 50% of the overall project. My specific tasks included:

* Task 1: Worked on half the rstudio code
* Task 2: worked on half the documentation

**Member 2: Yifan Chen**  
My teammate and I agree that I handled 50% of the overall project. My specific tasks included:

* Task 1: Worked on half the rstudio code
* Task 2: worked on half the documentation

Code:

library(sparklyr)

library(dplyr)

# connect to Spark using a local instance running on host computer

sc <- spark\_connect(master = "local")

# read data from host machine to R dataFrame. Tips: If read in arff file extension. The feature generated by DT will become wired.

getwd()

setwd("C:\\Users\\nicke\\Desktop\\New folder")

R\_df <- read.csv("PhishingData.csv", header=TRUE, sep=",")

# transfer R dataFrame into spark dataFrame, otherwise we cannot use ml\_decision\_tree() method

Sparkly\_df <- copy\_to(sc , R\_df)

# split data set into 70% training and 30% test, set seed to 1000 to add randomness when spliting

partitions <- Sparkly\_df %>% sdf\_partition(training = 0.7, test = 0.3, seed = 1000)

# set the formula for decision tree, Result, SFH, popUpWidow and other features

ml\_formula <- formula(Result ~ SFH + popUpWidnow + SSLfinal\_State + Request\_URL + URL\_of\_Anchor + web\_traffic + URL\_Length + age\_of\_domain + having\_IP\_Address)

# use training data to generate decision tree. Parameters: forula; type is classification, because in this question,

# we aim to classify if it's phishing WebSite. Not regression, it used for linear question; impurity is entropy,

# it depended on our type value, classification -> entropy; max.bins; max.depth; seed

model <- partitions$training %>% ml\_decision\_tree(ml\_formula, type = "classification", impurity = "entropy", max.bins = 16L, max.depth = 16L, seed = 42L)

# output this decision tree

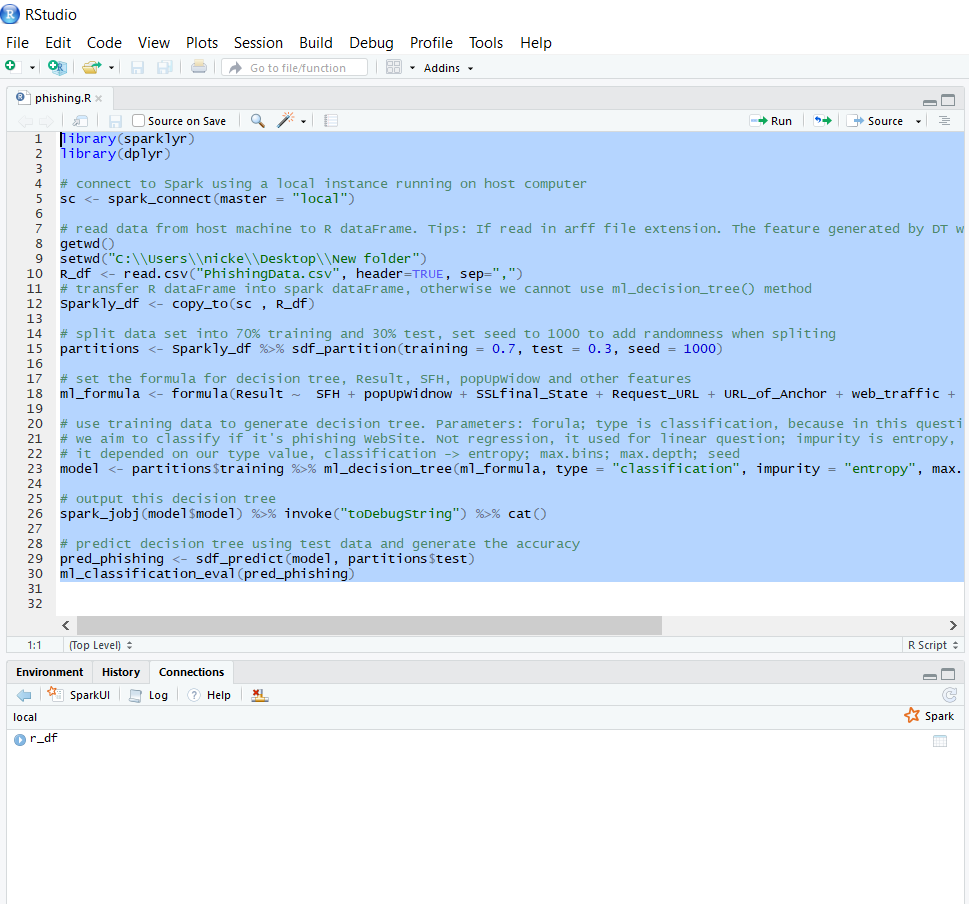
spark\_jobj(model$model) %>% invoke("toDebugString") %>% cat()

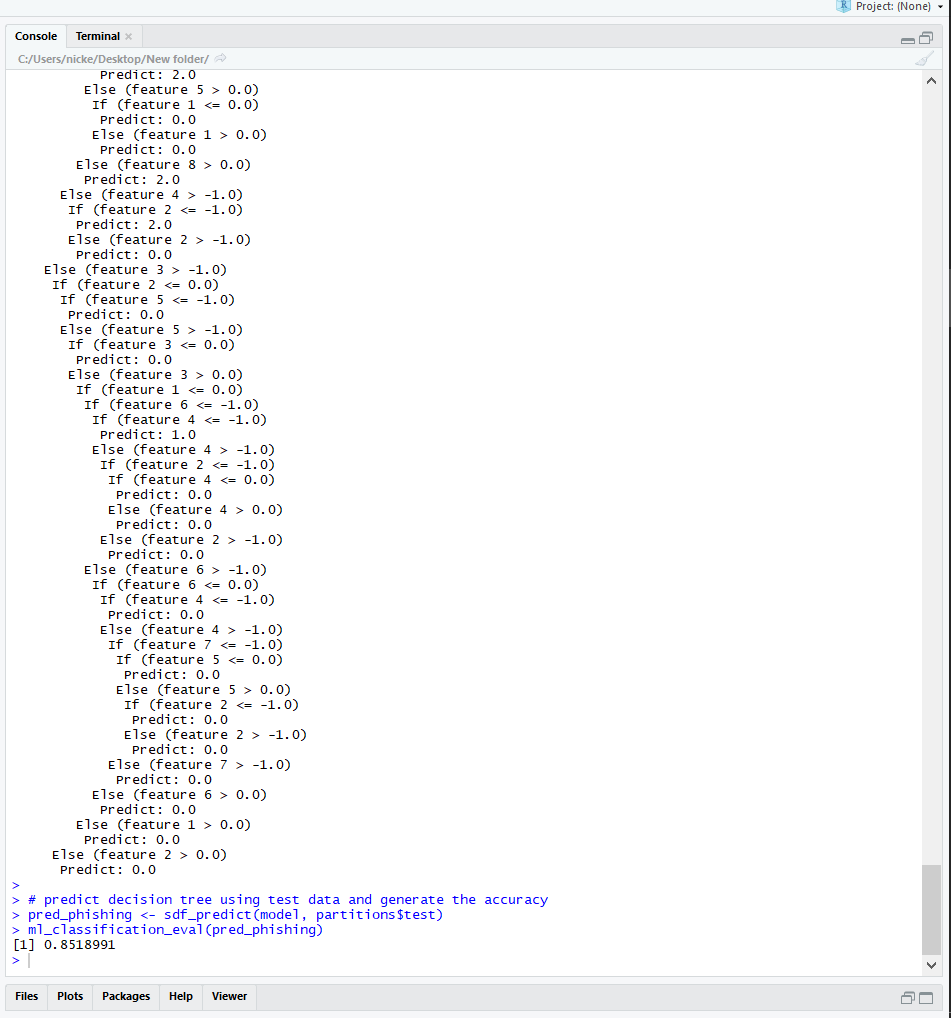
# predict decision tree using test data and generate the accuracy

pred\_phishing <- sdf\_predict(model, partitions$test)

ml\_classification\_eval(pred\_phishing)

Screenshots





**1. Use the 70%-30% ratio for training and testing**

# split data set into 70% training and 30% test, set seed to 1000 to add randomness when spliting

partitions <- Sparkly\_df %>% sdf\_partition(training = 0.7, test = 0.3, seed = 1000)

**2. Generate a decision tree which describes the relationship between the result and the nine categorical attributes**

# set the formula for decision tree, Result, SFH, popUpWidow and other features

ml\_formula <- formula(Result ~ SFH + popUpWidnow + SSLfinal\_State + Request\_URL + URL\_of\_Anchor + web\_traffic + URL\_Length + age\_of\_domain + having\_IP\_Address)

# use training data to generate decision tree. Parameters: forula; type is classification, because in this question,

# we aim to classify if it's phishing WebSite. Not regression, it used for linear question; impurity is entropy,

# it depended on our type value, classification -> entropy; max.bins; max.depth; seed

model <- partitions$training %>% ml\_decision\_tree(ml\_formula, type = "classification", impurity = "entropy", max.bins = 3L, max.depth = 16L, seed = 42L)

# output this decision tree

spark\_jobj(model$model) %>% invoke("toDebugString") %>% cat()

The tree is set up the results to look for popUpWidnow, SSLfinal\_State, Request\_URL, URL\_of\_Anchor, web\_traffic, URL\_Length, age\_of\_domain, having\_IP\_Address

**3. Report the classification accuracy of the learner on the training set**

# predict decision tree using test data and generate the accuracy

pred\_phishing <- sdf\_predict(model, partitions$test)

ml\_classification\_eval(pred\_phishing)

which returned [1] 0.8518991

Translated to %: 85.18991% accuracy for the model