Homework2

Narang,Mandeep

11/2/2018

installing required Packages

Here we need dplyr and ggplot packages to wrangle and visualize the dataset.

```
library(dplyr)
```

```
##
## 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
```

```
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

Data File is ## Reading the data in R from csv file

```
Sys_Data<- read.csv("SystemAdministrators.csv")
```

View Data

We can get the Idea about structure and contents of data file:

```
names(Sys_Data)
```

```
## [1] "Experience"     "Training"       "Completed.task"
```

```
glimpse(Sys_Data)
```

```
## Observations: 75
## Variables: 3
## $ Experience     <dbl> 10.9, 9.9, 10.4, 13.7, 9.4, 12.4, 7.9, 8.9, 10....
## $ Training       <int> 4, 4, 6, 6, 8, 4, 6, 4, 6, 4, 4, 4, 8, 4, 4, 4,...
## $ Completed.task <fct> Yes, Yes, Yes, Yes, Yes, Yes, Yes, Yes, Yes, Ye...
```

```
head(Sys_Data,10)
```

```
##    Experience Training Completed.task
## 1        10.9        4             Yes
## 2         9.9        4             Yes
## 3        10.4        6             Yes
## 4        13.7        6             Yes
## 5         9.4        8             Yes
## 6        12.4        4             Yes
## 7         7.9        6             Yes
## 8         8.9        4             Yes
## 9        10.2        6             Yes
## 10       11.4        4             Yes
```

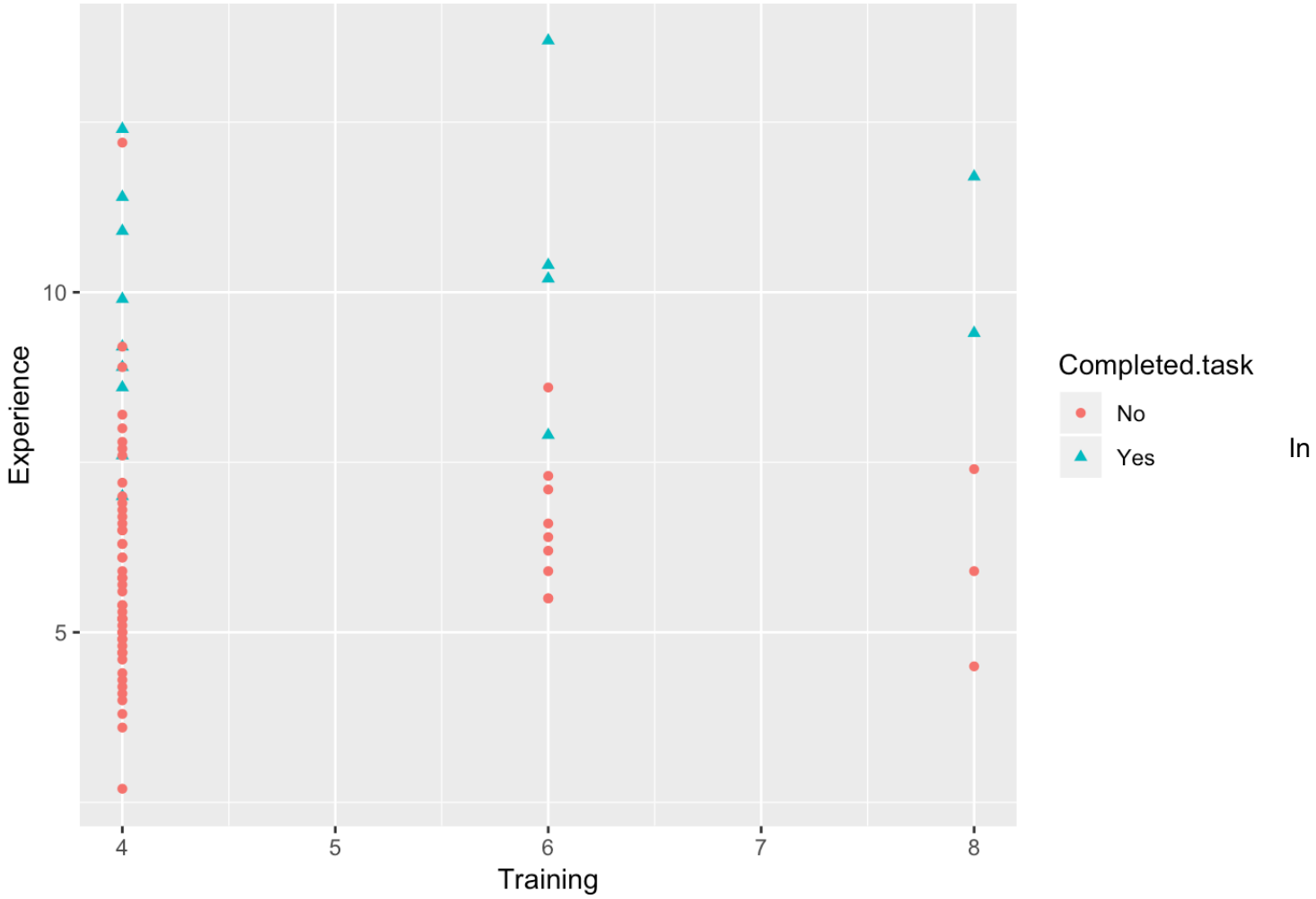
```
tail(Sys_Data,10)
```

```
##    Experience Training Completed.task
## 66        6.5        4              No
## 67        7.4        8              No
## 68        6.5        4              No
## 69        8.2        4              No
## 70        5.9        4              No
## 71        5.6        4              No
## 72        5.9        8              No
## 73        6.4        6              No
## 74        3.8        4              No
## 75        5.3        4              No
```

By running above command we have seen that data has three columns i.e.,“Experience”,“Training”,“Completed.task”. “Completed.task” is a factor with lables (Yes,No)

Q1:Using ggplot2 package, create a scatter plot of Experience vs. Training using color or symbol to distinguish programmers who completed the task from those who did not complete it. Which predictor(s) appear(s) potentially useful for classifying task completion?

```
Answer1<- ggplot(Sys_Data , aes(Training,Experience,color  = Completed.task , shape = Completed.task)) + geom_point()
Answer1
```



the above code a scatter plot is created with name of “Answer1”. having “Training” on X-axis and “Experience” on Y-axis using “Completed.task” to distinguish between Yes and No records using Different color and shape for them. As we can see in the above Answer1 plot “Experience” is a potential predictor in classifying the “Completed.task” field, because as Experience Increases we have more “Yes” in the data and

for a lower value of Experience we have more “No” records in the data. But If we have a look on “Training”, we can see that for same value of “Training” parameter we have similar records in “Yes” and “No”.

Q2.1 Run a logistic regression model with both predictors using the entire dataset as training data. Generate a confusion matrix and answer the following: among those who completed the task, what is the percentage of programmers incorrectly classified as failing to complete the task?

```
logit.reg <- glm(Completed.task ~ ., data = Sys_Data, family = "binomial")
logit.reg.pred <- predict(logit.reg, Sys_Data, type = "response")

summary(logit.reg)
```

```
##
## Call:
## glm(formula = Completed.task ~ ., family = "binomial", data = Sys_Data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.65306  -0.34959  -0.17479  -0.08196   2.21813
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -10.9813     2.8919  -3.797 0.000146 ***
## Experience    1.1269     0.2909   3.874 0.000107 ***
## Training      0.1805     0.3386   0.533 0.593970
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 75.060  on 74  degrees of freedom
## Residual deviance: 35.713  on 72  degrees of freedom
## AIC: 41.713
##
## Number of Fisher Scoring iterations: 6
```

```
levels(Sys_Data$Completed.task)<- c(FALSE ,TRUE)

Conf_matrix<-table(logit.reg.pred> 0.5,Sys_Data$Completed.task)
Conf_matrix
```

```
##
##         FALSE TRUE
##   FALSE    58    5
##   TRUE      2   10
```

In the Above code: Line 52 the model is trained by using entire data. line 53 predicted the values values for the same data by trained model. line 55 checked the summary of model and found the intercept and coffecicients and we can see that predictor “Training” is not statistically significant. Line 56 Created the confusion matrix named “Conf_matrix”

Q2.2:among those who completed the task, what is the percentage of programmers incorrectly classified as failing to complete the task?

```
Conf_matrix[1,2]/sum(Conf_matrix[,2])
```

```
## [1] 0.3333333
```

In the above confussion matrix we have total 15 “Yes/TRUE” records from the raw data and after predicting from the model, 5 them are classified as “No/FALSE”

Q3:How much experience must be accumulated by a programmer with 6 training credits before his or her estimated probability of completing the task exceeds 0.6? (Hint: in a logistic regression you can write the left hand-side of the regression equation as the log of odds).

```
Find_exp<- function(Training,p,model_logit) {
EXP<- (log(p/(1-p)) - coef(model_logit)["(Intercept)"] - coef(model_logit)["Training"
]*Training)/coef(model_logit)["Experience"]
return(EXP)
}

Experience<- Find_exp(6,0.6,logit.reg)
Experience
```

```
## (Intercept)
##    9.143164
```

In the above script a funcion has been made named “Find_exp” we can put Training,probability and model_logit to find Experience required.

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.