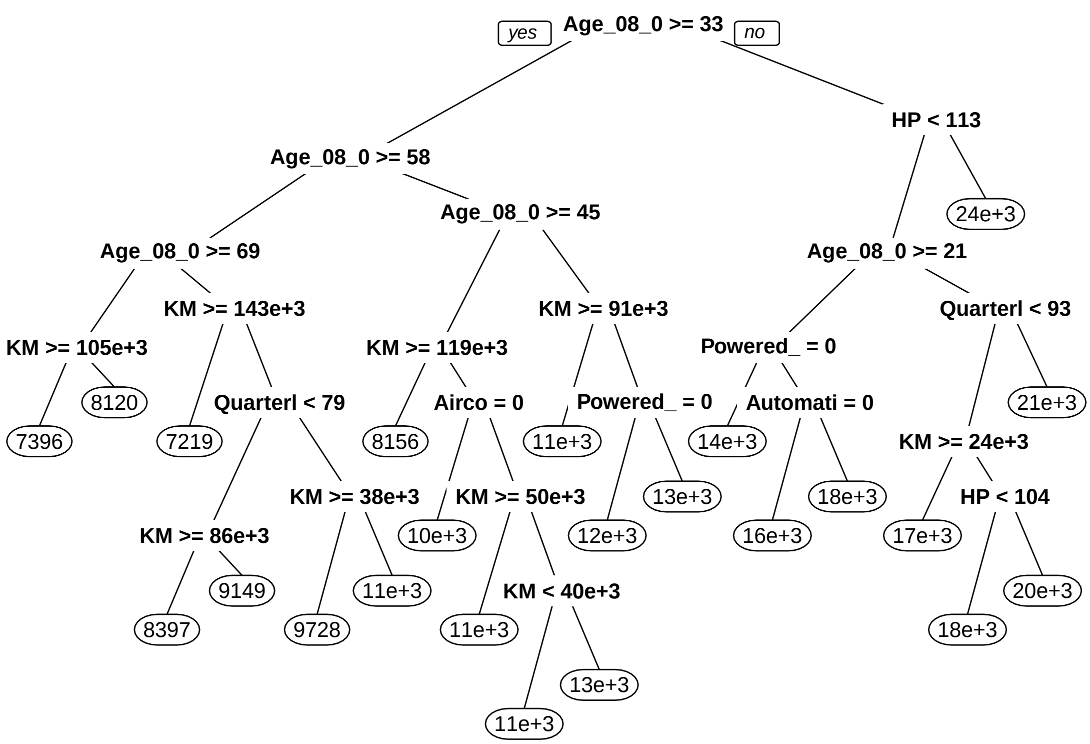
HW5

Hansika Karkera, Vrunda Shah

4/2/2020

#Problem 1

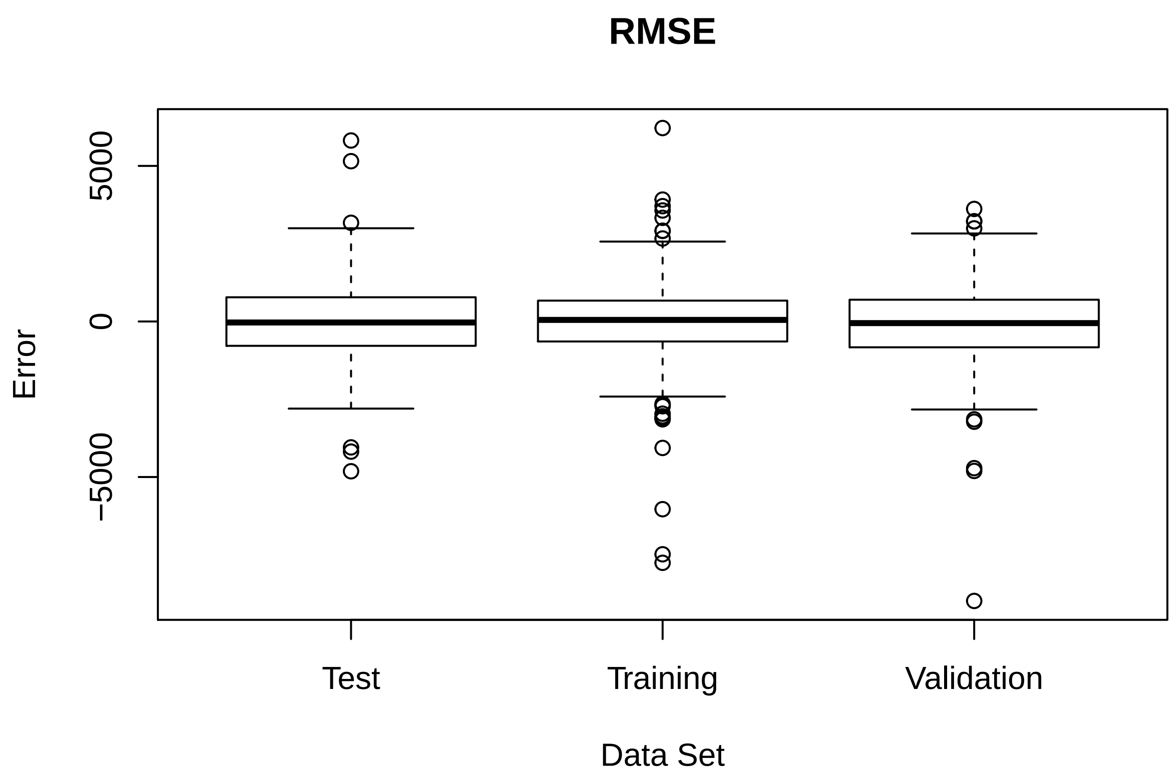
```r
# Problem 1(a)(i)
rf<- rpart(Price ~ Age_08_04+ KM + Fuel_Type + HP + Automatic + Doors + Quarterly_Tax + Mfr_Guarantee +
prp(rf)
```



The important predictors for predicting the car's price are - The age of the car, accumulated kilometers and horse power.

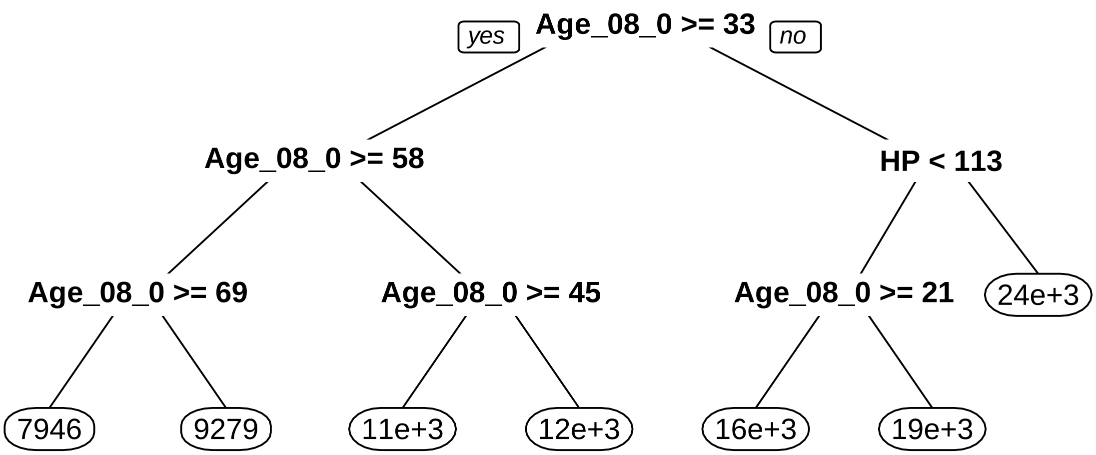
```r
# Problem 1(a)(ii)
# Root Mean Square Error for training, validation and test
rmse_train <- rmse(predict(rf, train_df[,]), train_df$Price)
rmse_valid <-rmse(predict(rf, valid_df[,]), valid_df$Price)
rmse_test <-rmse(predict(rf, test_df[,]), test_df$Price)
```

```
train_err <- predict(rf, train_df[,]) - train_df$Price
test_err <- predict(rf, test_df[,]) - test_df$Price
valid_err <-predict(rf, valid_df[,]) - valid_df$Price
# To create a box plot
  err <-data.frame(Error = c(train_err,test_err,valid_err),
                   Data = c(rep("Training", length(train_err)),
                          rep("Test", length(test_err)),
                          rep("Validation",length(valid_err))))
  boxplot(Error~Data, data=err, main="RMSE",
          xlab = "Data Set", ylab = "Error",border="black")
```



From the box plot we can observe that the test dataset has less number of outliers compared to the other two, meaning that there is a possibilty that the above model is underfit. This is probably possible because the training set data is not large enough.

```
#Problem 1(a)(iii)
rf_prune<- prune(rf,cp = 0.01) # cp selected from cptable
prp(rf_prune)
```




```
rmse_prune_train <- rmse(predict(rf_prune,train_df),train_df$Price)
rmse_prune_valid <- rmse(predict(rf_prune,valid_df),valid_df$Price)
rmse_prune_test <- rmse(predict(rf_prune,test_df),test_df$Price)
rmse <- data.frame("Data"=c("Training","Validation","Test"),"Prunned Tree RMSE"=c(rmse_prune_train,rmse_
rmse
```

```
##         Data Prunned.Tree.RMSE Full.Tree.RMSE
## 1   Training          1356.185       1131.551
## 2 Validation          1423.283       1252.984
## 3       Test          1470.345       1274.316
```

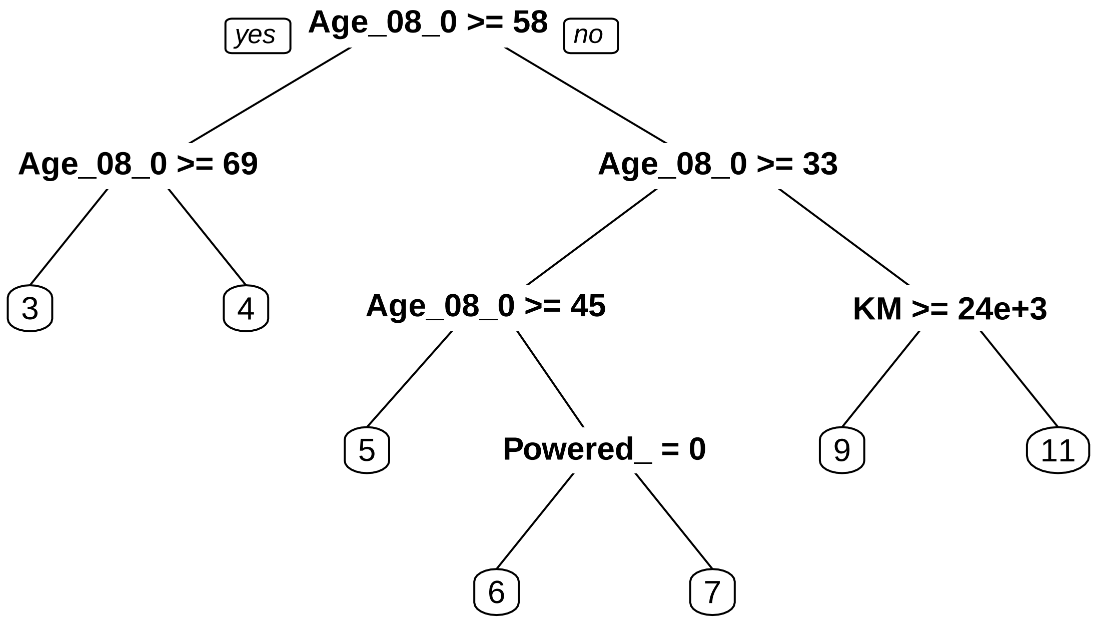
Compared to the prunned tree the the accuracy of the full tree is more for training set, validation and test set.

```
# Problem 1(b)
  bins <- seq(min(cars$Price),
              max(cars$Price),
              (max(cars$Price) - min(cars$Price))/20)
  bins
```

```
##  [1]  4350.0  5757.5  7165.0  8572.5  9980.0 11387.5 12795.0 14202.5 15610.0
## [10] 17017.5 18425.0 19832.5 21240.0 22647.5 24055.0 25462.5 26870.0 28277.5
## [19] 29685.0 31092.5 32500.0
```

```
  tags <- c(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20)
  BinnedPrice <- as.data.frame(cut(cars$Price,
                     breaks=bins,
                     include.lowest=TRUE,
                     right=FALSE,labels = tags))

  train_df$BinnedPrice<-BinnedPrice[train_set,]
  valid_df$BinnedPrice<-BinnedPrice[valid_set,]
  test_df$BinnedPrice<-BinnedPrice[test_set,]
# Classification Tree
ct<- rpart(BinnedPrice ~ Age_08_04+ KM + Fuel_Type + HP + Automatic + Doors + Quarterly_Tax + Mfr_Guara
prp(ct)
```



On comparing the two trees we observe that after creating bins the size of the tree has reduced and the there is a change in top variables affecting price as well.

```
# Problem 1(b)(ii)
new_data <- data.frame(Age_08_04=77,KM=117000,Fuel_Type="Petrol",HP=110,Automatic=0,Doors=5,Quarterly_T
predict_rt<- predict(rf,new_data)
predict_ct <- bins[predict(ct,new_data,type = "class")]
predict_rt
```

```
##        1
## 7395.714
```

```
predict_ct
```

```
## [1] 7165
```

Problem 1(b)(iii) Our prediction of the two models seem to have a differnce of less than $300. The full regression model returns a more accurate result compared to the classification model.Both models seem to be accurate but the regression model is better trained. The disadvatage of using decision tree is that they are prone to errors in classification, even a slight change in data, will change the entire model.

#Problem 2 Logit - -14.188 + 79.964*TotExp/Assets + 9.173*TotLns&Lses/Assets Odds - e^(-14.188 + 79.964*TotExp/Assets + 9.173*TotLns&Lses/Assets) Probabilities - 1 / 1 + e^(14.188 - 79.964*TotExp/Assets - 9.173*TotLns&Lses/Assets)

```
library(readxl)
bank_df <- read_excel("Banks.xlsx" , sheet = 1)
bank_df <- na.omit(bank_df)
bank_df$`Financial Condition`<- factor(bank_df$`Financial Condition`,levels=c(0,1), labels=c("Strong","

#Problem 2(A)1 creating logistic model
library(AER)
```

```
## Warning: package 'AER' was built under R version 3.6.3
```

```
## Loading required package: car
```

```
## Warning: package 'car' was built under R version 3.6.2
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## Loading required package: lmtest
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: sandwich
```

```
## Loading required package: survival
```

```r
fit2 <- glm(`Financial Condition` ~ `TotLns&Lses/Assets`+ `TotExp/Assets`, data = bank_df , family = "bi
summary(fit2)
```

```
##
## Call:
## glm(formula = `Financial Condition` ~ `TotLns&Lses/Assets` +
##     `TotExp/Assets`, family = "binomial", data = bank_df)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.64035  -0.35514   0.02079   0.53234   1.03373
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)           -14.188      6.122  -2.317   0.0205 *
## `TotLns&Lses/Assets`    9.173      6.864   1.336   0.1814
## `TotExp/Assets`        79.964     39.263   2.037   0.0417 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 27.726  on 19  degrees of freedom
## Residual deviance: 12.831  on 17  degrees of freedom
## AIC: 18.831
##
## Number of Fisher Scoring iterations: 6
```

```r
##Problem 2(A)2 odds as function of predictors
coef(fit2)
```

```
##          (Intercept) `TotLns&Lses/Assets`      `TotExp/Assets`
##           -14.187552             9.173215            79.963941
```

```r
odds <- exp(coef(fit2))
odds
```

```
##          (Intercept) `TotLns&Lses/Assets`      `TotExp/Assets`
##         6.893258e-07         9.635549e+03         5.344393e+34
```

```r
### Problem 2(A)3 probability as function of predictors
bank_df$prob <- predict(fit2 , newdata = bank_df , type = "response")
bank_df
```

```
## # A tibble: 20 x 6
##      Obs `Financial Cond~ `TotCap/Assets` `TotExp/Assets` `TotLns&Lses/As~
##    <dbl> <fct>                      <dbl>           <dbl>            <dbl>
##  1     1 Weak                         8.1            0.13             0.64
##  2     2 Weak                         6.6            0.1              1.04
##  3     3 Weak                         5.8            0.11             0.66
##  4     4 Weak                        12.3            0.09             0.8
```

```
## 5      5 Weak                        4.5            0.11              0.69
## 6      6 Weak                        9.1            0.14              0.74
## 7      7 Weak                        1.1            0.12              0.63
## 8      8 Weak                        8.9            0.12              0.75
## 9      9 Weak                        0.7            0.16              0.56
## 10    10 Weak                        9.8            0.12              0.65
## 11    11 Strong                      7.3            0.1               0.55
## 12    12 Strong                     14              0.08              0.46
## 13    13 Strong                      9.6            0.08              0.72
## 14    14 Strong                     12.4            0.08              0.43
## 15    15 Strong                     18.4            0.07              0.52
## 16    16 Strong                      8              0.08              0.54
## 17    17 Strong                     12.6            0.09              0.3
## 18    18 Strong                      9.8            0.07              0.67
## 19    19 Strong                      8.3            0.09              0.51
## 20    20 Strong                     20.6            0.13              0.79
## # ... with 1 more variable: prob <dbl>
```

```r
##Problem 2(B) creating new data
new_data <- data.frame(0.6,0.11 )
names(new_data)[1] <- "TotLns&Lses/Assets"
names(new_data)[2] <- "TotExp/Assets"
## calculating logit function
new_fit <- predict(fit2 , newdata = new_data , type = "response")
new_fit
```

```
##         1
## 0.5280731
```

```r
## calculating the odds
odd <- exp(new_fit)
odd
```

```
##        1
## 1.695662
```

```r
## calculating probability
prob <- predict(fit2, newdata=new_data, type="response")
prob
```

```
##         1
## 0.5280731
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 3.6.2
```

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

```
## Warning: package 'lattice' was built under R version 3.6.2
```

```
## Loading required package: ggplot2

##
## Attaching package: 'caret'

## The following object is masked from 'package:survival':
##
##     cluster

## The following objects are masked from 'package:Metrics':
##
##     precision, recall
```

```
fit3<- rpart(`Financial Condition` ~ `TotLns&Lses/Assets` + `TotExp/Assets`, data = bank_df, method = "c
pred3 <- predict(fit3, bank_df, type = "class")
confusionMatrix(pred3, bank_df$`Financial Condition`)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Strong Weak
##     Strong      7    0
##     Weak        3   10
##
##                Accuracy : 0.85
##                  95% CI : (0.6211, 0.9679)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 0.001288
##
##                   Kappa : 0.7
##
##  Mcnemar's Test P-Value : 0.248213
##
##             Sensitivity : 0.7000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.7692
##              Prevalence : 0.5000
##          Detection Rate : 0.3500
##    Detection Prevalence : 0.3500
##       Balanced Accuracy : 0.8500
##
##        'Positive' Class : Strong
##
```

```
#Problem 2(c)
cut_off_value<- as.numeric(0.5)
odds<- cut_off_value/(1- cut_off_value)
odds
```

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

```r
logit<-log(odds)
logit
```

```
## [1] 0
```

```r
#Problem 2(D)
TotLns.Lses.Assets <- 9.173215

TotExp.Assets <- 79.963941

Ratio <- TotLns.Lses.Assets/TotExp.Assets
Ratio
```

```
## [1] 0.1147169
```

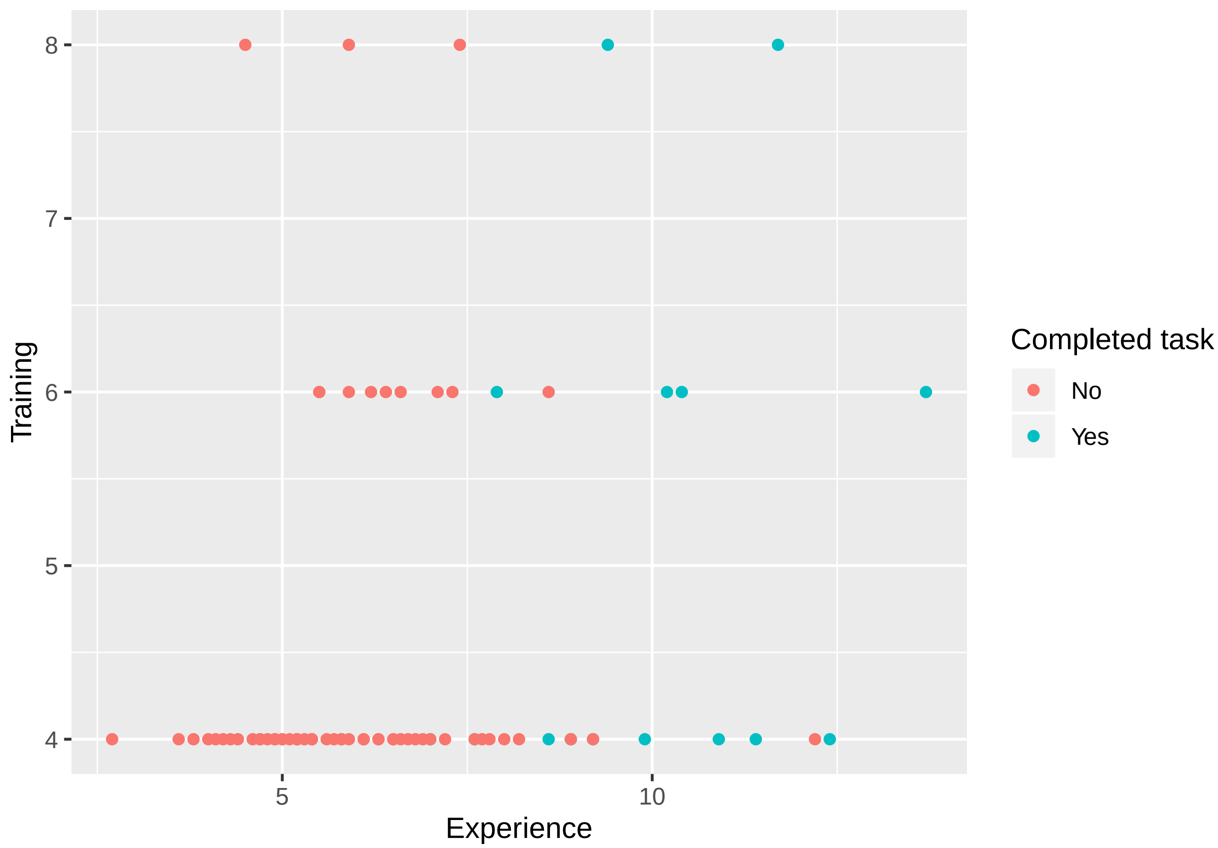
This ratio is classified as financially strong because ration is less than 0.5

##Problem 2(E) It is given that classification cost is much higher when a bank is declared strong but actually is weak. Therefore , we have to reduce entries being declared as strong. So, we decraese the cutoff value.

#Problem 3

```r
stu_df <- read_excel("System Administrators.xlsx" , sheet = 1)
stu_df <- na.omit(stu_df)
test_df <- stu_df
stu_df$Complete <- 1* (stu_df$`Completed task` == "Yes")
stu_df <- stu_df[,-c(3)]

## Problem 3(a)
p <- ggplot(test_df, aes(x = Experience, y = Training, colour = `Completed task`)) +
  geom_point() + xlab("Experience") + ylab("Training")
p
```




```r
#Problem 3(B)
## creating the model
 fit1 <- glm(Complete ~., data = stu_df, family = "binomial")
data.frame(summary(fit1)$coefficients, odds = exp(coef(fit1)))
```

```
##                Estimate Std..Error    z.value     Pr...z..         odds
## (Intercept) -10.9813061  2.8919380 -3.7972135 0.0001463318 1.701686e-05
## Experience    1.1269310  0.2908785  3.8742325 0.0001069613 3.086170e+00
## Training      0.1805094  0.3386087  0.5330913 0.5939704002 1.197827e+00
```

```r
round(data.frame(summary(fit1)$coefficients, odds = exp(coef(fit1))),5)
```

```
##               Estimate Std..Error  z.value Pr...z..    odds
## (Intercept) -10.98131    2.89194 -3.79721  0.00015 0.00002
## Experience    1.12693    0.29088  3.87423  0.00011 3.08617
## Training      0.18051    0.33861  0.53309  0.59397 1.19783
```

```r
summary(fit1)
```

```
##
## Call:
## glm(formula = Complete ~ ., family = "binomial", data = stu_df)
##
## 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
```

```r
## creating confusion matrix
table(ifelse(fit1$fitted > 0.5, 1, 0), stu_df$Complete)
```

```
##
##      0  1
##   0 58  5
##   1  2 10
```

Total completed task <- 15 Incorrectly classified <- 5

```r
percentage <- 5/15 * 100
percentage
```

```
## [1] 33.33333
```

The percentage is 33.33%

##Problem 3(c) To decrease the percentage in part(b) the cutoff probability should be increased

```r
##Problem 3(D)
summary(fit1)
```

```
##
## Call:
## glm(formula = Complete ~ ., family = "binomial", data = stu_df)
##
## 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
```

```r
#intercept
b0 <- -10.98131
# coeeficient Expereince
b1 <- 1.12693
# coeeficient Training
b2 <- 0.18051
```

So , p <- 1/(1+e^-(bo+b1x1+b2x2)) here b0, b1 , b2 are given x2 <- 4 (given) p <- 0.5 solving the equation and subtituting the v alue of each variable we get x1 <- 9.11 x1 ~ 9