**Reading the file**

> data <- read.csv("E:\\ TYCS\\Data Science\\logistic regression\\diabetes.csv",head=TRUE,sep=",")

**Importing the library**

> library(caTools)

**Splitting dataset for training and testing purpose**

> split <- sample.split(data,SplitRatio=0.8)

> split

[1] TRUE TRUE TRUE TRUE TRUE FALSE TRUE FALSE TRUE

> training <- subset(data , split=="TRUE")

> testing <- subset(data , split=="FALSE")

**Creating a model**

> model <- glm(outcome~.,training,family="binomial")

**Summary of the dataset**

> summary(model)

Call:

glm(formula = outcome ~ ., family = "binomial", data = training)

Coefficients:

Estimate Std. Error z value Pr(>|z|)

(Intercept) -7.9943481 0.8208528 -9.739 < 2e-16 \*\*\*

Sr.No -0.0004125 0.0004706 -0.877 0.3807

npreg 0.1503180 0.0379618 3.960 7.50e-05 \*\*\*

glu 0.0348409 0.0037995 9.170 < 2e-16 \*\*\*

bp -0.0118886 0.0059568 -1.996 0.0460 \*

skin -0.0038409 0.0070696 -0.543 0.5869

bmi 0.0877451 0.0168294 5.214 1.85e-07 \*\*\*

ped 0.7137638 0.3293002 2.168 0.0302 \*

age 0.0091117 0.0103356 0.882 0.3780

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 787.83 on 597 degrees of freedom

Residual deviance: 568.55 on 589 degrees of freedom

AIC: 586.55

Number of Fisher Scoring iterations: 5

**Making predictions**

> res <- predict(model,training,type="response")

> res

1 2 3 4 5 7 9 10

0.715967741 0.051544271 0.837216611 0.055114932 0.867888195 0.082261451 0.807386853 0.050956074

11 12 13 14 16 18 19 20

0.278891671 0.924477966 0.789358908 0.801591619 0.451344563 0.261090336 0.369306309 0.252020729

21 22 23 25 27 28 29 30

0.457879460 0.374826679 0.952346887 0.776308001 0.785367103 0.060966724 0.644305215 0.333232075

31 32 34 36 37 38 39 40

0.412376903 0.632220537 0.054197128 0.173172401 0.746537726 0.408921892 0.169812995 0.544631054

41 43 45 46 47 48 49 50

0.780372224 0.137917548 0.689264826 0.930075205 0.457708526 0.043102159 0.420080953 0.053754754

52 54 55 56 57 58 59 61

0.094250028 0.873931904 0.778215466 0.027382609 0.920873040 0.324043298 0.789020553 0.012322793

63 64 65 66 67 68 70 72

0.028409615 0.335962208 0.416579253 0.139060909 0.171147613 0.456120522 0.369897389 0.429204739

73 74 75 76 77 79 81 82

0.864901617 0.360301798 0.055281359 0.002340241 0.097272669 0.680796775 0.124887982 0.007569387

83 84 85 86 88 90 91 92

0.159048790 0.054053278 0.729589105 0.214054076 0.213490438 0.085264456 0.024204001 0.327293592

93 94 95 97 99 100 101 102

0.360630060 0.312943509 0.283399239 0.092056613 0.165098602 0.493706829 0.822352264 0.327377417

103 104 106 108 109 110 111 112

0.077959053 0.037861244 0.270204107 0.478226649 0.116305858 0.103424847 0.661172497 0.819865023

113 115 117 118 119 120 121 122

0.060673251 0.780978088 0.410017702 0.183488911 0.133174263 0.071208906 0.872019000 0.314117176

124 126 127 128 129 130 131 133

0.369828613 0.517227512 0.499485661 0.214900765 0.217566173 0.136296140 0.702405797 0.726389153

135 136 137 138 139 140 142 144

0.062216927 0.294471014 0.101668950 0.074334591 0.268032505 0.268320501 0.335822104 0.461920302

145 146 147 148 149 151 153 154

0.577673034 0.007531056 0.076615636 0.274556328 0.644502719 0.388265836 0.863142643 0.633974375

155 156 157 158 160 162 163 164

0.969516890 0.883908358 0.101515666 0.136738337 0.977264937 0.338764359 0.283362183 0.112402225

165 166 167 169 171 172 173 174

0.278925993 0.274113904 0.474623619 0.265916811 0.193876574 0.584684244 0.181401021 0.178347119

175 176 178 180 181 182 183 184

0.055458354 0.887485846 0.784920873 0.665472210 0.062984718 0.276077591 0.002374041 0.062990598

185 187 189 190 191 192 193 194

0.383155083 0.892580111 0.281290945 0.424201089 0.098306216 0.524812763 0.731485811 0.974636342

0.214343598 0.257699162 0.811897101 0.116740310 0.159898922 0.103671889 0.167738646 0.195055742

741 742 743 745 747 748 749 750

0.759602612 0.158339738 0.095004467 0.927614581 0.627577873 0.225967938 0.826263239 0.554824064

751 752 754 756 757 758 759 760

0.496272929 0.240108448 0.727183688 0.354816178 0.433809155 0.262223763 0.154525650 0.884408239

761 763 765 766 767 768

0.076431898 0.098309569 0.273516670 0.176515456 0.237709816 0.058966011

> (table(ActualValue=training$outcome,PredictedValue=res>0.3))

PredictedValue

ActualValue FALSE TRUE

0 266 111

1 40 181

**Calculating the accuracy**

> (297+156)/(297+99+45+156)

[1] 0.758794

**Importing the library**

> library(ROCR)

> ROCRpred = prediction(res, training$outcome)

> ROCRpref = performance(ROCRpred,"tpr" , "fpr")

**Plotting the graph**

> plot(ROCRpref,colorize=TRUE,print.cuttofs.at=seq(0.1,by=0.5))

