# CIS 530- ADVANCED DATA MINING

**(Spring 2024)**

DIABETES PREDICTION SYSTEM

## Professor:

**Name: Prof. Hua Fang**

**Email:** [**hfang@umassd.edu**](mailto:hfang@umassd.edu)

## By:-

JAI SAXENA

Student ID: 02129249

Introduction

Diabetes is threatening a lot of people nowadays, without having a perfect cure for it. There are two types of diabetes, namely Type-1 and Type-2. Type-2 diabetes is commonly called diabetes mellitus. It can be defined as a chronic condition that affects the way the body processes blood sugar (glucose).

We consider the mellitus here. After deep research we found that, that some parameters are directly responsible for the mellitus to occur. By using the data of the people with and without diabetes, a dataset has been built. We use that dataset to classify the people who are at risk of getting diabetes.

Loading the required libraries

library(ggplot2)

library(ggvis)

library(corrplot)

library(caTools)

library(ROCR)

Data Loading

The observations of the people are stored in a CSV format, named ‘diabetes.csv’. The data is loaded in the environment. Let’s check how the data is structured.

data = read.csv("C:/Users/crsri/Documents/Diabetes\_Prediction/Data/diabetes.csv")

head(data)

## Pregnancies Glucose BloodPressure SkinThickness Insulin BMI

## 1 6 148 72 35 0 33.6

## 2 1 85 66 29 0 26.6

## 3 8 183 64 0 0 23.3

## 4 1 89 66 23 94 28.1

## 5 0 137 40 35 168 43.1

## 6 5 116 74 0 0 25.6

## DiabetesPedigreeFunction Age Outcome

## 1 0.627 50 1

## 2 0.351 31 0

## 3 0.672 32 1

## 4 0.167 21 0

## 5 2.288 33 1

## 6 0.201 30 0

summary(data)

## Pregnancies Glucose BloodPressure SkinThickness

## Min. : 0.000 Min. : 0.0 Min. : 0.00 Min. : 0.00

## 1st Qu.: 1.000 1st Qu.: 99.0 1st Qu.: 62.00 1st Qu.: 0.00

## Median : 3.000 Median :117.0 Median : 72.00 Median :23.00

## Mean : 3.845 Mean :120.9 Mean : 69.11 Mean :20.54

## 3rd Qu.: 6.000 3rd Qu.:140.2 3rd Qu.: 80.00 3rd Qu.:32.00

## Max. :17.000 Max. :199.0 Max. :122.00 Max. :99.00

## Insulin BMI DiabetesPedigreeFunction Age

## Min. : 0.0 Min. : 0.00 Min. :0.0780 Min. :21.00

## 1st Qu.: 0.0 1st Qu.:27.30 1st Qu.:0.2437 1st Qu.:24.00

## Median : 30.5 Median :32.00 Median :0.3725 Median :29.00

## Mean : 79.8 Mean :31.99 Mean :0.4719 Mean :33.24

## 3rd Qu.:127.2 3rd Qu.:36.60 3rd Qu.:0.6262 3rd Qu.:41.00

## Max. :846.0 Max. :67.10 Max. :2.4200 Max. :81.00

## Outcome

## Min. :0.000

## 1st Qu.:0.000

## Median :0.000

## Mean :0.349

## 3rd Qu.:1.000

## Max. :1.000

str(data)

## 'data.frame': 768 obs. of 9 variables:

## $ Pregnancies : int 6 1 8 1 0 5 3 10 2 8 ...

## $ Glucose : int 148 85 183 89 137 116 78 115 197 125 ...

## $ BloodPressure : int 72 66 64 66 40 74 50 0 70 96 ...

## $ SkinThickness : int 35 29 0 23 35 0 32 0 45 0 ...

## $ Insulin : int 0 0 0 94 168 0 88 0 543 0 ...

## $ BMI : num 33.6 26.6 23.3 28.1 43.1 25.6 31 35.3 30.5 0 ...

## $ DiabetesPedigreeFunction: num 0.627 0.351 0.672 0.167 2.288 ...

## $ Age : int 50 31 32 21 33 30 26 29 53 54 ...

## $ Outcome : int 1 0 1 0 1 0 1 0 1 1 ...

Correlations

The proportionalities of the attributes of the data can be identified by the correlation coefficient, either numerically or visually. They helps to know which attributes are highly dependent on the prediction variable: Outcome.

correlations <- cor(data)

correlations

## help Pregnancies Glucose BloodPressure

## Pregnancies 1.00000000 0.12945867 0.14128198

## Glucose 0.12945867 1.00000000 0.15258959

## BloodPressure 0.14128198 0.15258959 1.00000000

## SkinThickness -0.08167177 0.05732789 0.20737054

## Insulin -0.07353461 0.33135711 0.08893338

## BMI 0.01768309 0.22107107 0.28180529

## DiabetesPedigreeFunction -0.03352267 0.13733730 0.04126495

## Age 0.54434123 0.26351432 0.23952795

## Outcome 0.22189815 0.46658140 0.06506836

## SkinThickness Insulin BMI

## Pregnancies -0.08167177 -0.07353461 0.01768309

## Glucose 0.05732789 0.33135711 0.22107107

## BloodPressure 0.20737054 0.08893338 0.28180529

## SkinThickness 1.00000000 0.43678257 0.39257320

## Insulin 0.43678257 1.00000000 0.19785906

## BMI 0.39257320 0.19785906 1.00000000

## DiabetesPedigreeFunction 0.18392757 0.18507093 0.14064695

## Age -0.11397026 -0.04216295 0.03624187

## Outcome 0.07475223 0.13054795 0.29269466

## DiabetesPedigreeFunction Age Outcome

## Pregnancies -0.03352267 0.54434123 0.22189815

## Glucose 0.13733730 0.26351432 0.46658140

## BloodPressure 0.04126495 0.23952795 0.06506836

## SkinThickness 0.18392757 -0.11397026 0.07475223

## Insulin 0.18507093 -0.04216295 0.13054795

## BMI 0.14064695 0.03624187 0.29269466

## DiabetesPedigreeFunction 1.00000000 0.03356131 0.17384407

## Age 0.03356131 1.00000000 0.23835598

## Outcome 0.17384407 0.23835598 1.00000000

corrplot(correlations, method="color")

A blue and white squares with red text

Description automatically generated

Visualization

Visualizations are used to grasp the structure of data and its relations, like how they vary and their relationships with the other data. They are said to be EDA (Exploratory Data Analysis).

A matrix of scatterplots is produced for this dataset.

pairs(data, col=data$Outcome)

A screenshot of a chart

Description automatically generated

Glucose and Insulin

The glucose and the insulin are the major factors of the diabetes…which in turn have direct proportionality in the future during the diabetes. They are the major cause of the occurrence. They are strong correlated on each other.

data %>% ggvis(~Glucose,~Insulin,fill =~Outcome) %>% layer\_points()

020406080100120140160180200Glucose0100200300400500600700800InsulinOutcome0.01.0

BMI and Diabetes Pedigree Function

The BMI and Diabetes Pedigree Function is plotted here.

data %>% ggvis(~BMI,~DiabetesPedigreeFunction,fill =~Outcome) %>% layer\_points()

010203040506070BMI0.00.20.40.60.81.01.21.41.61.82.02.22.4DiabetesPedigreeFunctionOutcome0.01.0

Age and Pregnancies

The males have 0 for the pregnancy attribute, which is why we find a lot of values plotting zero in this graph.

data %>% ggvis(~Age,~Pregnancies,fill =~Outcome) %>% layer\_points()

20253035404550556065707580Age0246810121416PregnanciesOutcome0.01.0

Preparing the data

The dataset is divided as two parts, training data and testing data, with a Split ratio of 0.75. It means that 2/3rds of the data is labelled by training set and the rest 1/3rd of data is the testing set. The division of the dataset is by means of a random order generated by the seed.

set.seed(88)

split <- sample.split(data$Outcome, SplitRatio = 0.75)

data\_train <- subset(data, split == TRUE)

data\_test <- subset(data, split == FALSE)

Logistic regression

Logistic regression helps to classify whether the concern person will get diabetes or not. Since we are using logistic regression, we have to mention that family = binomial. We are using all the attributes we have in the dataset. Let us take a look at the summary.

model <- glm (Outcome ~ .-Pregnancies + Glucose + BloodPressure + SkinThickness + Insulin + BMI + DiabetesPedigreeFunction + Age, data = data\_train, family = binomial)

summary(model)

##

## Call:

## glm(formula = Outcome ~ . - Pregnancies + Glucose + BloodPressure +

## SkinThickness + Insulin + BMI + DiabetesPedigreeFunction +

## Age, family = binomial, data = data\_train)

##

## Deviance Residuals:

## Min 1Q Median 3Q Max

## -2.4254 -0.7250 -0.4361 0.7487 2.9829

##

## Coefficients:

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

## (Intercept) -8.3339721 0.8159489 -10.214 < 2e-16 \*\*\*

## Glucose 0.0382162 0.0044235 8.639 < 2e-16 \*\*\*

## BloodPressure -0.0088309 0.0060059 -1.470 0.1415

## SkinThickness 0.0007624 0.0081902 0.093 0.9258

## Insulin -0.0017095 0.0010823 -1.580 0.1142

## BMI 0.0792632 0.0169318 4.681 2.85e-06 \*\*\*

## DiabetesPedigreeFunction 0.7386714 0.3332368 2.217 0.0266 \*

## Age 0.0204344 0.0095270 2.145 0.0320 \*

## ---

## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

##

## (Dispersion parameter for binomial family taken to be 1)

##

## Null deviance: 745.11 on 575 degrees of freedom

## Residual deviance: 552.82 on 568 degrees of freedom

## AIC: 568.82

##

## Number of Fisher Scoring iterations: 5

Prediction

The trained model is used to predict the data for the testing data and for the training data (For checking accuracy purposes and for ROC curve)

predict\_train <- predict(model, type = 'response')

predict\_test <- predict(model, newdata = data\_test, type = 'response')

ROC Curve

ROCRpred <- prediction(predict\_train, data\_train$Outcome)

ROCRperf <- performance(ROCRpred, 'tpr','fpr')

plot(ROCRperf, colorize = TRUE, text.adj = c(-0.2,1.7))

A graph with a line

Description automatically generated with medium confidence

Comparison

By comparing the real values with the real data, we can see how our machine learning algorithm performs.

predict\_test\_c = predict\_test

i = 1

**while**(i <= length(predict\_test))

{

**if**(predict\_test[i] < 0.5)

predict\_test\_c[i] = 0

**else**

predict\_test\_c[i] = 1

i = i + 1;

}

compare <- data.frame(data\_test$Outcome,predict\_test\_c)

colnames(compare) <- c("Observed Values","Predicted values")

ggplot(data = compare,aes(x = "Observed Values", y = "Predicted values")) + geom\_abline() +

xlab("Observed Values") + ylab("Predicted values") + theme\_classic()

A black line on a white background

Description automatically generated

compare

## Observed Values Predicted values

## 6 0 0

## 8 0 0

## 9 1 1

## 10 1 0

## 11 0 0

## 14 1 1

## 16 1 0

## 32 1 1

## 33 0 0

## 37 0 0

## 38 1 0

## 41 0 1

## 44 1 1

## 45 0 1

## 46 1 1

## 49 1 0

## 56 0 0

## 60 0 0

## 62 1 0

## 70 0 0

## 73 1 1

## 77 0 0

## 78 0 0

## 84 0 0

## 86 0 0

## 89 1 1

## 91 0 0

## 93 0 0

## 94 1 0

## 95 0 0

## 102 0 0

## 103 0 0

## 105 0 0

## 110 1 0

## 111 1 1

## 114 0 0

## 124 0 0

## 128 0 0

## 130 1 0

## 142 0 0

## 143 0 0

## 150 0 0

## 153 1 1

## 163 0 0

## 164 0 0

## 168 0 0

## 182 0 0

## 192 0 0

## 195 0 0

## 199 1 0

## 201 0 0

## 204 0 0

## 209 0 0

## 216 1 1

## 219 1 0

## 225 0 0

## 227 0 0

## 228 1 1

## 236 1 1

## 239 1 1

## 240 0 0

## 244 1 0

## 256 1 0

## 262 1 1

## 264 0 1

## 272 0 0

## 280 0 0

## 281 1 1

## 283 0 0

## 285 1 0

## 291 0 0

## 292 1 0

## 299 1 0

## 304 1 1

## 312 0 0

## 315 1 0

## 323 1 0

## 326 0 0

## 327 1 0

## 341 0 0

## 342 0 0

## 343 0 0

## 344 0 0

## 346 0 0

## 350 1 0

## 356 1 1

## 357 1 0

## 358 1 1

## 363 0 0

## 364 1 1

## 367 1 0

## 374 0 0

## 379 1 1

## 381 0 0

## 382 0 0

## 388 1 0

## 391 0 0

## 392 1 1

## 395 1 1

## 396 0 0

## 408 0 0

## 414 0 0

## 417 0 0

## 419 0 0

## 422 0 0

## 424 0 0

## 431 0 0

## 432 0 0

## 433 0 0

## 436 1 1

## 437 0 1

## 439 0 0

## 448 0 0

## 449 1 0

## 450 0 0

## 451 0 0

## 453 0 0

## 456 1 1

## 463 0 0

## 466 0 0

## 473 0 0

## 478 0 0

## 493 0 0

## 498 0 0

## 500 0 1

## 504 0 0

## 508 0 0

## 509 0 0

## 513 0 0

## 531 0 0

## 532 0 0

## 533 0 0

## 536 1 1

## 538 0 0

## 542 1 0

## 543 1 0

## 548 0 0

## 550 0 1

## 562 1 1

## 563 0 0

## 567 0 0

## 573 0 0

## 577 0 0

## 580 1 1

## 583 0 0

## 585 1 0

## 586 0 0

## 592 0 0

## 599 1 1

## 606 0 0

## 608 0 0

## 610 0 0

## 623 0 1

## 625 0 0

## 627 0 0

## 636 1 0

## 639 1 0

## 640 0 0

## 652 0 0

## 655 0 0

## 663 1 1

## 664 1 1

## 665 1 0

## 671 0 1

## 673 0 0

## 675 0 0

## 680 0 0

## 681 0 0

## 691 0 0

## 694 1 1

## 695 0 0

## 700 0 1

## 703 1 1

## 711 0 0

## 714 0 0

## 719 0 0

## 721 0 0

## 724 0 0

## 728 0 0

## 736 0 0

## 740 1 0

## 741 1 1

## 744 1 1

## 746 0 0

## 747 1 1

## 749 1 1

## 753 0 0

## 757 0 0

## 759 0 0

## 760 1 1

## 764 0 0

## 766 0 0

Conclusion

The results can be improved more by applying more feature scaling and data cleaning. From this project we predicted Type-2 diabetes, commonly called diabetes mellitus.

As a result, it can help to improve their health conditions.