

Diagnose Diabetes

UPx academy certification exam

Manoj Kalamkar | Data Science Super Specialization | June 10th, 2017

# Problem Statement

The dataset consists data of patients, all are females at least 21 years old of Pima Indian heritage.

**Attribute Information:**

1. Number of times pregnant

2. Plasma glucose concentration a 2 hour in an oral glucose tolerance test

3. Diastolic blood pressure (mm Hg)

4. Triceps skin fold thickness (mm)

5. 2-Hour serum insulin (mu U/ml)

6. Body mass index (weight in kg/ (height in m) ^2)

7. Diabetes pedigree function

8. Age (years)

9. Diabetes (0 or 1)

***Objective***

Classify positive and negative cases by splitting your dataset into training and testing, also implement cross validation to avoid overfitting. Evaluate your model based on AUC curve.

# Exploratory Data Analysis

## Understanding Data

> dim(df\_diabetes)

[1] 768 9

> str(df\_diabetes)

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

$ times\_pregnant: int 6 1 8 1 0 5 3 10 2 8 ...

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

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

$ triceps : 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

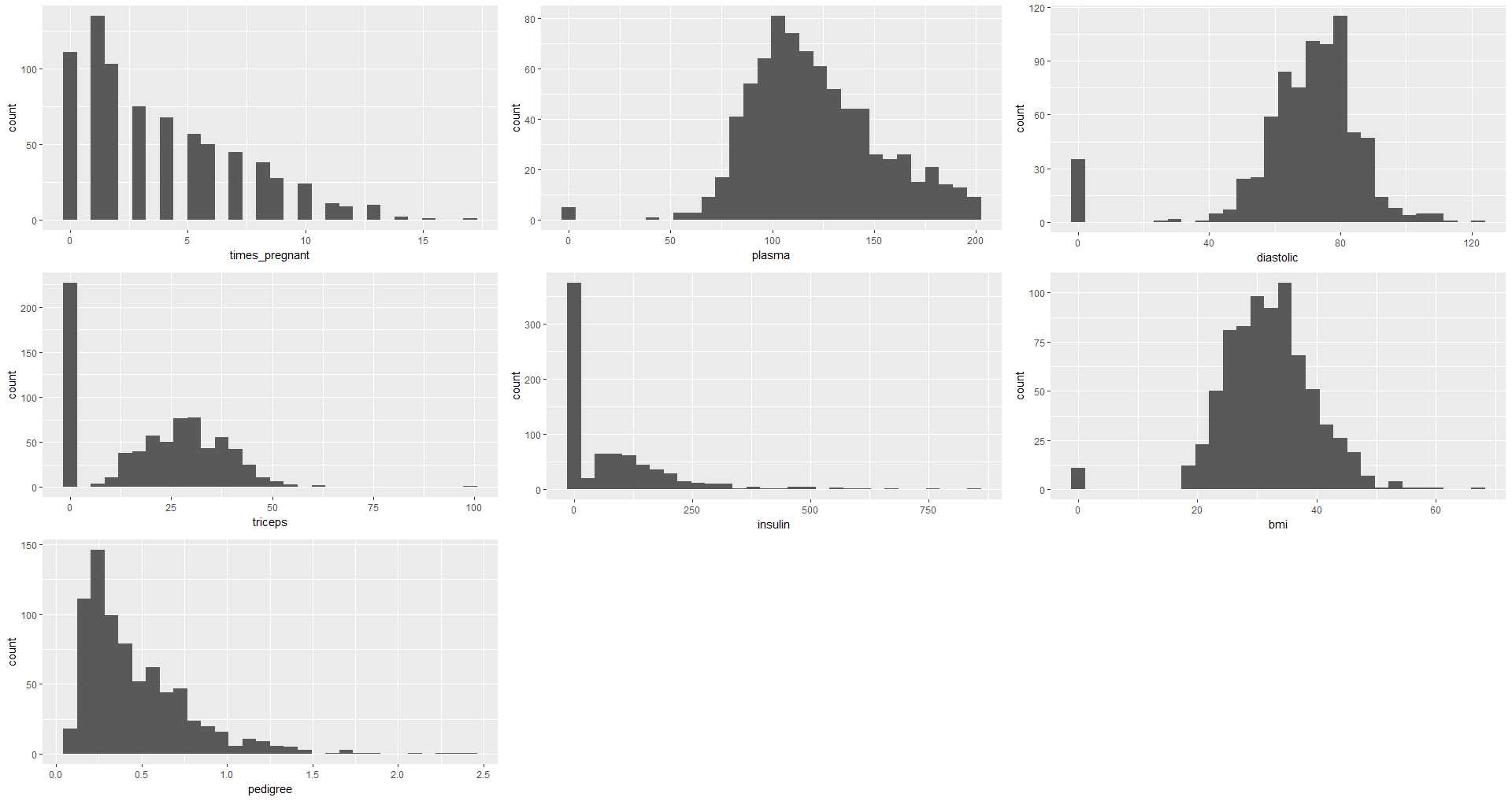
$ pedigree : num 0.627 0.351 0.672 0.167 2.288 ...

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

$ diabetes : int 1 0 1 0 1 0 1 0 1 1 ...

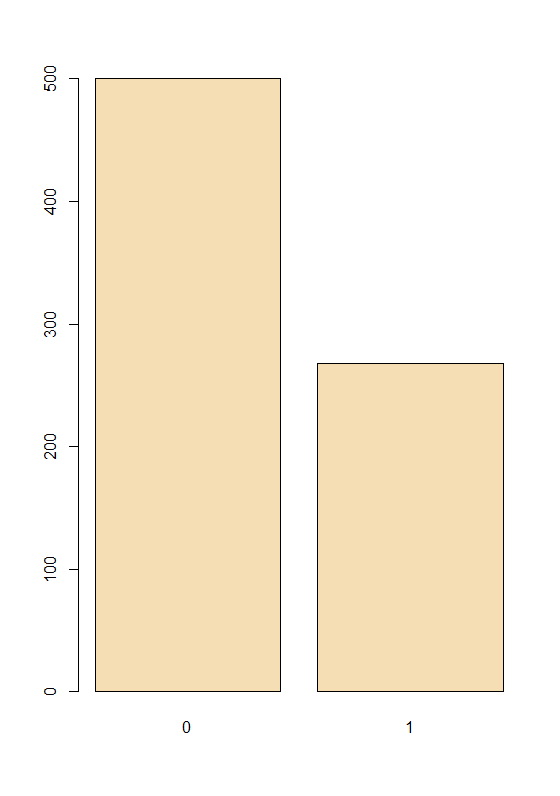
‘diabetes’ feature has only 2 possible values and ideally should be a factor datatype

## Univariate Analysis



Observations:-

Not much of a pattern seen



Observations:-

Looks like 50% of patients had diabetes as compared to patients who don’t have.

## Bivariate Analysis

Observations:-

Plasma, Time Pregnant, Diastolic, Triceps seem to have good predictors.

Rest are bit scattered

# Feature Engineering



Observations:-

Seems to be no missing values

# Build Predictive Models

> summary(results)

Call:

summary.resamples(object = results)

Models: m\_lda, m\_glm, m\_cart, m\_knn, m\_svm, m\_rf, m\_gbm

Number of resamples: 10

Accuracy

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

m\_lda 0.7273 0.7435 0.7727 0.7761 0.7993 0.8312 0

m\_glm 0.6753 0.7672 0.7792 0.7707 0.7922 0.8182 0

m\_cart 0.7143 0.7273 0.7468 0.7552 0.7857 0.8158 0

m\_knn 0.6623 0.7143 0.7386 0.7371 0.7727 0.8026 0

m\_svm 0.7273 0.7427 0.7713 0.7669 0.7922 0.8052 0

m\_rf 0.7013 0.7557 0.7727 0.7683 0.7915 0.8182 0

m\_gbm 0.7143 0.7305 0.7647 0.7670 0.7890 0.8421 0

Kappa

Min. 1st Qu. Median Mean 3rd Qu. Max. NA's

m\_lda 0.3631 0.4258 0.4857 0.4820 0.5222 0.6128 0

m\_glm 0.2418 0.4741 0.4936 0.4682 0.5255 0.5867 0

m\_cart 0.3392 0.3621 0.4053 0.4300 0.5030 0.5668 0

m\_knn 0.2043 0.3296 0.3891 0.3910 0.4672 0.5403 0

m\_svm 0.3631 0.4071 0.4551 0.4563 0.5116 0.5437 0

m\_rf 0.3149 0.4456 0.5031 0.4753 0.5243 0.5867 0

m\_gbm 0.3388 0.3775 0.4502 0.4664 0.5192 0.6358 0

Observations:-

At the moment GLM, LDA and Random Forest seems to be standing out as better models

# Predict Models with validation set

confusionMatrix(predict\_rf,test\_diabetes$diabetes)

Confusion Matrix and Statistics

Reference

Prediction 0 1

0 100 0

1 0 53

Accuracy : 1

95% CI : (0.9762, 1)

No Information Rate : 0.6536

P-Value [Acc > NIR] : < 2.2e-16

Kappa : 1

Mcnemar's Test P-Value : NA

Sensitivity : 1.0000

Specificity : 1.0000

Pos Pred Value : 1.0000

Neg Pred Value : 1.0000

Prevalence : 0.6536

Detection Rate : 0.6536

Detection Prevalence : 0.6536

Balanced Accuracy : 1.0000

'Positive' Class : 0

Observations:-

Random Forest is giving 100% accuracy and selected as best model for this set of data

# AUC Curve



Observations:-

Another proof of Random Forest being perfect model for this data