Diabetics Prediction using Logistic Regression and K-Nearest Neighbors

student

Abstract

Diabetes is quite possibly the most widely recognized human illness overall and may cause a few wellbeing related entanglements. It is answerable for significant bleakness, mortality, and financial misfortune. A convenient conclusion and forecast of this sickness could furnish patients with a chance to take the suitable preventive and therapy systems. To work on the comprehension of chance variables, we foresee diabetes for Pima Indian ladies using a strategic relapse model and K-Nearest Neighbors AI calculation. Our examination tracks down five fundamental indicators of diabetes: glucose, pregnancy, weight list (BMI), diabetes family capacity, and pulse . We further investigate a K-Nearest Neighbors to supplement and approve our examination. for this situation we checked the responsiveness out. Our favored particular yields an expectation awareness of 58.92% in strategic regression and a 50% sensitivity of KNN closest neighbors. We contend that our model can be applied to make a sensible expectation of diabetes, and might actually be utilized to supplement existing preventive measures to control the frequency of diabetes and lessen related costs.

Introduction

Diabetes is quite possibly the most widely recognized human sickness and has turned into a huge general wellbeing concern around the world. There were roughly 450 million individuals determined to have diabetes that came about in around 1.37 million passings universally in 2017 . In excess of 100 million US grown-ups live with diabetes, and diabetes was the seventh driving reason for death in the US in 2020 . One out of ten US grown-ups have diabetes now, and assuming the latest thing proceeds, it is projected that upwards of one out of three US grown-ups could have diabetes by 2050 . Diabetes patients are at raised hazard of creating unexpected issues like kidney disappointment, vision misfortune, coronary illness, stroke, sudden passing, and removal of feet or legs, which can prompt brokenness and constant harm of tissue . What's more, there are significant monetary expenses related with the sickness. The complete assessed cost of analyzed diabetes in the US expanded to USD 237 billion out of 2017 from USD 188 billion of every 2012. The overabundance clinical expenses per individual related with diabetes expanded to USD 9601 from USD 8417 during a similar period . Furthermore, there could be efficiency misfortune because of diabetic patients in the labor force.

A person at high gamble of diabetes may not know about the gamble factors related with it. Given the high predominance and seriousness of diabetes, specialists are keen on observing the most well-known risk variables of diabetes, as it very well may be because of a blend of a few reasons. Deciding the gamble factors and early expectation of diabetes have been imperative in decreasing diabetes confusions and financial weight and is helpful from both clinical practice and general wellbeing viewpoints . Also, investigations discover that screening high-risk people recognizes the populace bunches in which carrying out measures pointed toward forestalling diabetes will be the most helpful . Early intercession might assist with forestalling inconveniences and work on personal satisfaction and are fundamental in planning successful anticipation procedures . There is developing proof that way of life alteration forestalls or defers diabetes . The principal risk variables of diabetes are viewed as an unfortunate eating regimen, maturing, family ancestry, ethnic gatherings, heftiness, inactive way of life, and past history of gestational diabetes . Past examinations have additionally revealed that sex, weight list (BMI), pregnancy, and metabolic status are related with diabetes .

Literature Review

Forecast models can screen pre-diabetes or individuals with an expanded gamble of creating diabetes to assist with choosing the best clinical administration for patients. Various prescient conditions have been proposed

to demonstrate the gamble variables of occurrence diabetes . For example, Heikes et al. concentrated on an instrument to anticipate the gamble of diabetes in the US utilizing undiscovered and pre-diabetes information, and Razavian et al. created strategic relapse based forecast models for type 2 diabetes event. These models likewise assist with screening people to place people who are at a high gamble of having diabetes. Zou et al. utilized AI techniques to anticipate diabetes in Luzhou, China, and a five-overlay cross-approval was utilized to approve the models. Nguyen et al. [5] foresee the beginning of diabetes utilizing profound learning calculations recommending that refined techniques might work on the exhibition of models. Conversely, a few different examinations have shown that calculated relapse proceeds as least as well as AI strategies for infection risk forecast , for instance). Additionally, Anderson et al.(utilized calculated relapse alongside AI calculations and observed a higher precision with the strategic relapse model. These are predominantly founded on surveying risk elements of diabetes, like family and individual qualities; notwithstanding, the absence of a goal and fair assessment is as yet an issue . Furthermore, there is developing worry that those prescient models are inadequately evolved because of unseemly choice of covariates, missing information, little examples size, and wrongly determined measurable models . To this end, a couple of chance forecast models have been regularly utilized in clinical practice. The dependability and nature of these prescient devices and conditions show critical variety relying upon geology, accessible information, and identity . Risk factors for one ethnic gathering may not be summed up to other people; for instance, the commonness of diabetes is accounted for to be higher among the Pima Indian people group. Thus, this study utilizes the Pima Indian dataset to foresee assuming an individual is in danger of creating diabetes in light of explicit demonstrative elements

The objective of this task is to assemble a calculated relapse model that would foresee the probability of diabetes.This dataset is initially from the National Institute of Diabetes and Digestive and Kidney Diseases. The target of the dataset is to analytically anticipate whether a patient has diabetes, in view of specific demonstrative estimations remembered for the dataset. A few imperatives were put on the determination of these cases from a bigger data set. Specifically, all patients here are females something like 21 years of age of Pima Indian legacy.

Library and Setup

```
#Load required packages
library(car)
```

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

```
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
library(class)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## 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(tidyr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.6     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x purrr::lift()   masks caret::lift()
## x dplyr::recode() masks car::recode()
## x purrr::some()   masks car::some()
```

```
library(performance)
```

Methodology

1.Logistic Regression #data preparation

```
library(readxl)
diabetes  <-  read_excel("C:/Users/Admin/Downloads/may work/pima.xlsx")
view(diabetes)
glimpse(diabetes)
```

```
## Rows: 768
## Columns: 9
## $ Pregnancies              <dbl> 6, 1, 8, 1, 0, 5, 3, 10, 2, 8, 4, 10, 10, 1, ~
## $ Glucose                  <dbl> 148, 85, 183, 89, 137, 116, 78, 115, 197, 125~
## $ BloodPressure            <dbl> 72, 66, 64, 66, 40, 74, 50, 0, 70, 96, 92, 74~
## $ SkinThickness            <dbl> 35, 29, 0, 23, 35, 0, 32, 0, 45, 0, 0, 0, 0, ~
## $ Insulin                  <dbl> 0, 0, 0, 94, 168, 0, 88, 0, 543, 0, 0, 0, 0, ~
## $ BMI                      <dbl> 33.6, 26.6, 23.3, 28.1, 43.1, 25.6, 31.0, 35.~
## $ DiabetesPedigreeFunction <dbl> 0.627, 0.351, 0.672, 0.167, 2.288, 0.201, 0.2~
## $ Age                      <dbl> 50, 31, 32, 21, 33, 30, 26, 29, 53, 54, 30, 3~
## $ Outcome                  <dbl> 1, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, ~
```

Then we change the variable type according to what it should be

```
diabetes <- diabetes %>%
  mutate(Outcome = factor(Outcome, levels = c(0,1), labels = c("Not Diabetes", "Diabetes")))
glimpse(diabetes)
```

```
## Rows: 768
## Columns: 9
## $ Pregnancies              <dbl> 6, 1, 8, 1, 0, 5, 3, 10, 2, 8, 4, 10, 10, 1, ~
## $ Glucose                  <dbl> 148, 85, 183, 89, 137, 116, 78, 115, 197, 125~
## $ BloodPressure            <dbl> 72, 66, 64, 66, 40, 74, 50, 0, 70, 96, 92, 74~
## $ SkinThickness            <dbl> 35, 29, 0, 23, 35, 0, 32, 0, 45, 0, 0, 0, 0, ~
## $ Insulin                  <dbl> 0, 0, 0, 94, 168, 0, 88, 0, 543, 0, 0, 0, 0, ~
## $ BMI                      <dbl> 33.6, 26.6, 23.3, 28.1, 43.1, 25.6, 31.0, 35.~
## $ DiabetesPedigreeFunction <dbl> 0.627, 0.351, 0.672, 0.167, 2.288, 0.201, 0.2~
## $ Age                      <dbl> 50, 31, 32, 21, 33, 30, 26, 29, 53, 54, 30, 3~
## $ Outcome                  <fct> Diabetes, Not Diabetes, Diabetes, Not Diabete~
```

Pregnancies - Number of times pregnant.

Glucose - Plasma glucose concentration (glucose tolerance test).

BloodPressure - Diastolic blood pressure (mm Hg).

SkinThickness - Triceps skin fold thickness (mm).

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

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

DiabetesPedigreeFunction - Diabetes pedigree function.

Age - Age (years).

Outcome - Test for Diabetes

Exploratory Data Analysis

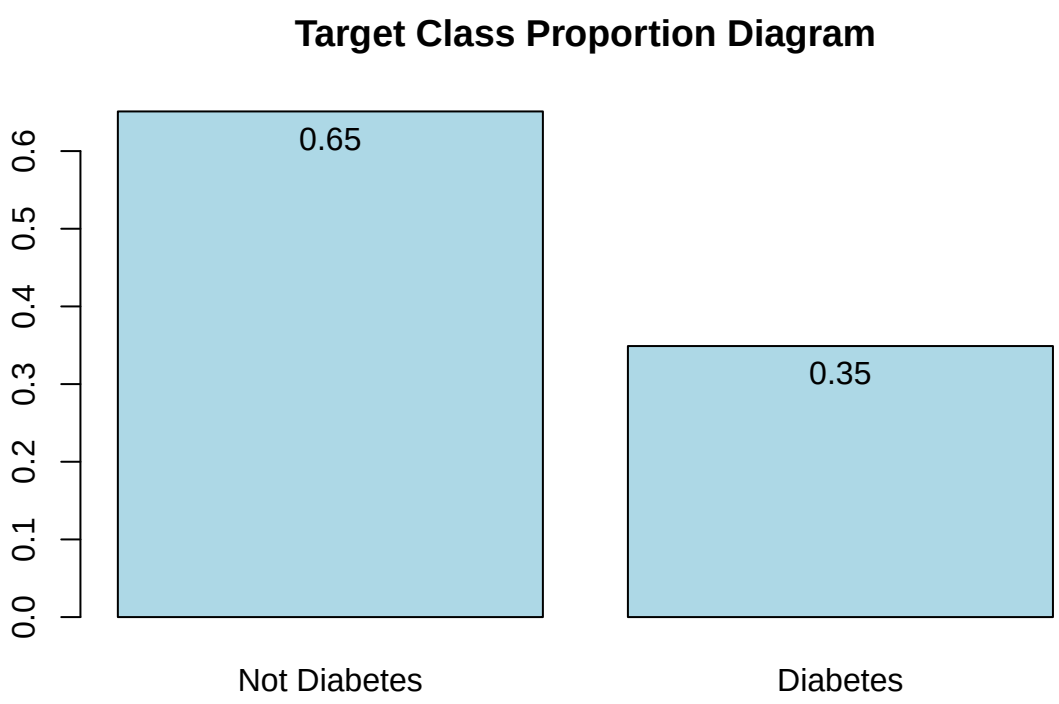
Checking for missing values.

```
colSums(is.na(diabetes))
```

```
##              Pregnancies                  Glucose            BloodPressure
##                        0                        0                        0
##            SkinThickness                  Insulin                      BMI
##                        0                        0                        0
## DiabetesPedigreeFunction                      Age                  Outcome
##                        0                        0                        0
```

We really want to check the class extent of the objective variable.

```
x <- prop.table(table(diabetes$Outcome))
b <- barplot(x,col="lightBlue", main = "Target Class Proportion Diagram")
text(x=b, y= x, labels=round(x,2), pos = 1)
```



The objective variable class is still somewhat adjusted

Cross Validation

Before we make the model, we really want to divide the information into train dataset and test dataset. We will utilize 80% of the information as the preparation information and the remainder of it as the testing information.

```
set.seed(23)

intrain <- sample(nrow(diabetes),nrow(diabetes)*.8)

diabetes_train <- diabetes[intrain,]
diabetes_test <- diabetes[-intrain,]
glimpse(diabetes_train)
```

```
## Rows: 614
## Columns: 9
## $ Pregnancies              <dbl> 7, 0, 7, 14, 0, 0, 11, 3, 5, 4, 2, 1, 3, 10, ~
## $ Glucose                  <dbl> 161, 165, 195, 175, 162, 95, 138, 89, 147, 15~
## $ BloodPressure            <dbl> 86, 90, 70, 62, 76, 64, 74, 74, 78, 62, 64, 7~
## $ SkinThickness            <dbl> 0, 33, 33, 30, 56, 39, 26, 16, 0, 31, 23, 21,~
## $ Insulin                  <dbl> 0, 680, 145, 0, 100, 105, 144, 85, 0, 284, 0,~
## $ BMI                      <dbl> 30.4, 52.3, 25.1, 33.6, 53.2, 44.6, 36.1, 30.~
## $ DiabetesPedigreeFunction <dbl> 0.165, 0.427, 0.163, 0.212, 0.759, 0.366, 0.5~
## $ Age                      <dbl> 47, 23, 55, 38, 25, 22, 50, 38, 65, 23, 21, 2~
## $ Outcome                  <fct> Diabetes, Not Diabetes, Diabetes, Diabetes, D~
```

We want to check again the extent of our train dataset, wheter it is as yet adjusted or not.

```
prop.table(table(diabetes_train$Outcome))
```

```
##
## Not Diabetes     Diabetes
##    0.6547231    0.3452769
```

Modelling

We will attempt to make a few models the Logistic Regression involving Outcome as the objective worth. The models that we will make come from multiple ways, some from the my comprehension or assessment and from stepwise determination.

```
LR_diabetes_model_all <- glm(formula = Outcome ~ ., data = diabetes_train, family = "binomial")
LR_diabetes_model_none <- glm(formula = Outcome~1,data = diabetes_train,family = "binomial")
summary(LR_diabetes_model_all)
```

```
##
## Call:
## glm(formula = Outcome ~ ., family = "binomial", data = diabetes_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4360  -0.7095  -0.4133   0.7038   2.9869
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -8.2000112  0.7862022 -10.430  < 2e-16 ***
## Pregnancies               0.1454703  0.0357046   4.074 4.62e-05 ***
## Glucose                   0.0346899  0.0041664   8.326  < 2e-16 ***
## BloodPressure            -0.0106300  0.0058804  -1.808 0.070654 .
## SkinThickness            -0.0017115  0.0074996  -0.228 0.819485
## Insulin                  -0.0004724  0.0010022  -0.471 0.637390
## BMI                       0.0750515  0.0164221   4.570 4.87e-06 ***
## DiabetesPedigreeFunction  1.2795020  0.3477698   3.679 0.000234 ***
## Age                       0.0110541  0.0103337   1.070 0.284749
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 791.41  on 613  degrees of freedom
## Residual deviance: 572.80  on 605  degrees of freedom
## AIC: 590.8
##
## Number of Fisher Scoring iterations: 5
```

```
summary(LR_diabetes_model_none)
```

```
##
## Call:
```

```
## glm(formula = Outcome ~ 1, family = "binomial", data = diabetes_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9204  -0.9204  -0.9204   1.4584   1.4584
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.63987    0.08488  -7.539 4.75e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 791.41  on 613  degrees of freedom
## Residual deviance: 791.41  on 613  degrees of freedom
## AIC: 793.41
##
## Number of Fisher Scoring iterations: 4
```

```
LR_diabetes_model_selected <- glm(formula = Outcome~Pregnancies+Glucose+BloodPressure+BMI+Insulin+Diabe
summary(LR_diabetes_model_selected)
```

```
##
## Call:
## glm(formula = Outcome ~ Pregnancies + Glucose + BloodPressure +
##     BMI + Insulin + DiabetesPedigreeFunction, family = "binomial",
##     data = diabetes_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4991  -0.7201  -0.4183   0.6936   3.0344
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -8.0082532  0.7598779 -10.539  < 2e-16 ***
## Pregnancies               0.1653155  0.0310173   5.330 9.83e-08 ***
## Glucose                   0.0358716  0.0040403   8.878  < 2e-16 ***
## BloodPressure            -0.0098997  0.0057463  -1.723 0.084925 .
## BMI                       0.0716483  0.0155500   4.608 4.07e-06 ***
## Insulin                  -0.0006439  0.0009051  -0.711 0.476811
## DiabetesPedigreeFunction  1.2785414  0.3451298   3.705 0.000212 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 791.41  on 613  degrees of freedom
## Residual deviance: 574.02  on 607  degrees of freedom
## AIC: 588.02
##
## Number of Fisher Scoring iterations: 5
```

```
LR_diabetes_model_backward <- step(object = LR_diabetes_model_all,direction = "backward",trace = F)
summary(LR_diabetes_model_backward)
```

```
##
## Call:
## glm(formula = Outcome ~ Pregnancies + Glucose + BloodPressure +
##     BMI + DiabetesPedigreeFunction, family = "binomial", data = diabetes_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5003  -0.7076  -0.4180   0.6782   3.0206
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -7.907660   0.743416 -10.637  < 2e-16 ***
## Pregnancies               0.167674   0.030831   5.438 5.37e-08 ***
## Glucose                   0.035055   0.003854   9.096  < 2e-16 ***
## BloodPressure            -0.010003   0.005732  -1.745 0.080978 .
## BMI                       0.070367   0.015403   4.569 4.91e-06 ***
## DiabetesPedigreeFunction  1.250997   0.341748   3.661 0.000252 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 791.41  on 613  degrees of freedom
## Residual deviance: 574.52  on 608  degrees of freedom
## AIC: 586.52
##
## Number of Fisher Scoring iterations: 5
```

After making several models, now let's compare each other.

```
compare_performance(LR_diabetes_model_all,LR_diabetes_model_backward,LR_diabetes_model_selected)
```

```
## # Comparison of Model Performance Indices
##
## Name                       | Model |     AIC | AIC weights |     BIC | BIC weights | Tjur's R2 |  RMS
## ------------------------------------------------------------------------------------------------------
## LR_diabetes_model_all      |   glm | 590.798 |       0.074 | 630.578 |     < 0.001 |     0.330 | 0.38
## LR_diabetes_model_backward |   glm | 586.524 |       0.628 | 613.044 |       0.951 |     0.328 | 0.38
## LR_diabetes_model_selected |   glm | 588.021 |       0.297 | 618.961 |       0.049 |     0.329 | 0.38
```

In picking which model is awesome, the AIC worth can be thought of. AIC attempts to gauge how much "data misfortune" starting with one model then onto the next. The more modest the AIC, the less data is lost, so the better the model is in anticipating the information. What's more, from the outcome above, we will involve LR_diabetes_model_backward as our Logistic Regression model.

```
summary(LR_diabetes_model_backward)
```

```
##
```

```
## Call:
## glm(formula = Outcome ~ Pregnancies + Glucose + BloodPressure +
##     BMI + DiabetesPedigreeFunction, family = "binomial", data = diabetes_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.5003  -0.7076  -0.4180   0.6782   3.0206
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -7.907660   0.743416 -10.637  < 2e-16 ***
## Pregnancies                0.167674   0.030831   5.438 5.37e-08 ***
## Glucose                    0.035055   0.003854   9.096  < 2e-16 ***
## BloodPressure             -0.010003   0.005732  -1.745 0.080978 .
## BMI                        0.070367   0.015403   4.569 4.91e-06 ***
## DiabetesPedigreeFunction   1.250997   0.341748   3.661 0.000252 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 791.41  on 613  degrees of freedom
## Residual deviance: 574.52  on 608  degrees of freedom
## AIC: 586.52
##
## Number of Fisher Scoring iterations: 5
```

LR_diabetes_model_backward's outline above contains part of data. In any case, we really want more spotlight on Pr(>|t|) and AIC. From Pr(>|t|) above, we can get data on which indicators affect the objective, assuming the worth is beneath 0.05 (alpha), we asume that the variable has tremendous impact toward the model, and afterward the more modest the Pr(>|t|) esteem, the more critical the indicators have on the objective, and to make it simpler, there is a star image which shows the more stars the more huge the indicator's effect on the objective.

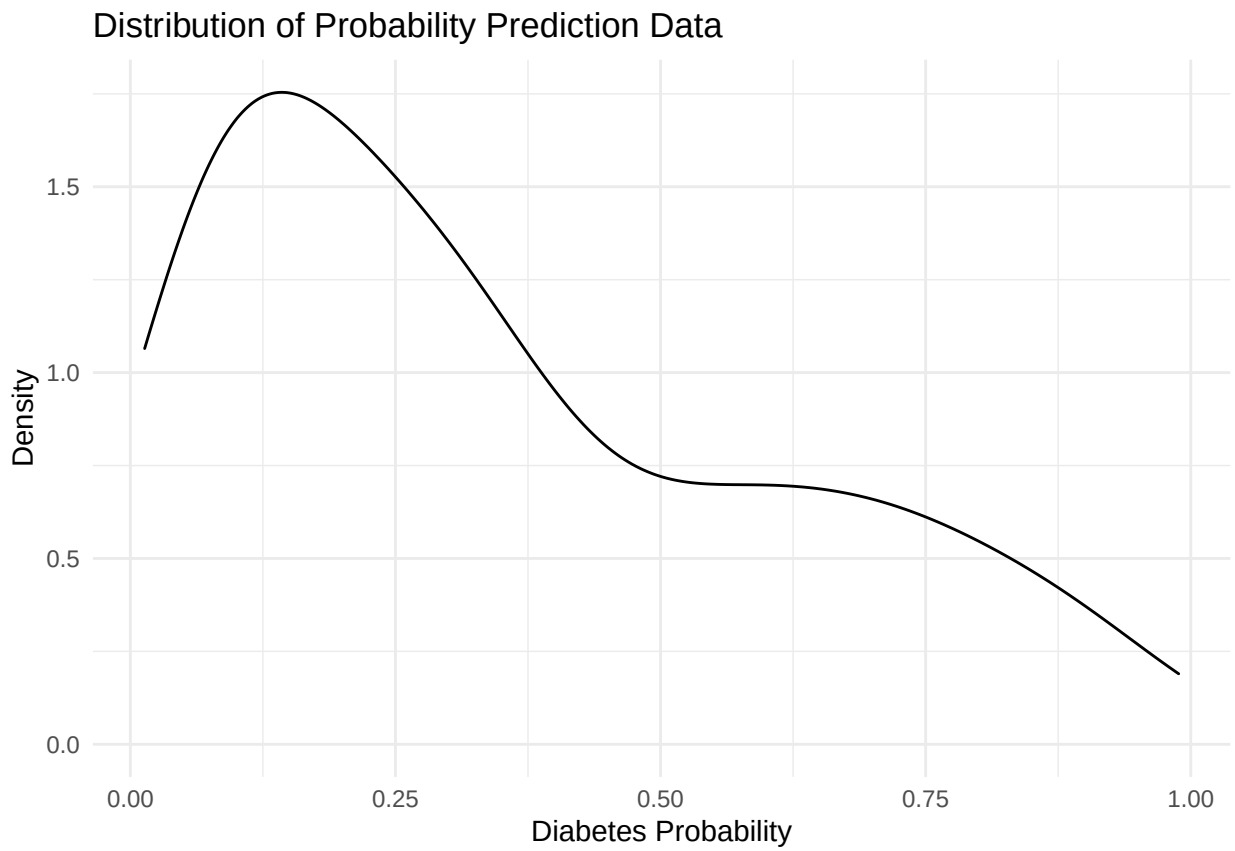
prediciting

In the wake of picking the best model for our dataset, then, at that point, we really want to test our model presentation utilizing testing dataset that we have splitted above.

```
diabetes_test_LR <- diabetes_test
diabetes_test_LR$pred_model_backward <- predict(object = LR_diabetes_model_backward,newdata = diabetes_
glimpse(diabetes_test_LR$label_model_backward<-as.factor(ifelse(diabetes_test_LR$pred_model_backward >
```

```
##  Factor w/ 2 levels "Diabetes","Not Diabetes": 2 1 2 2 2 2 2 1 2 2 ...
##  - attr(*, "names")= chr [1:154] "1" "2" "3" "4" ...
```

```
ggplot(diabetes_test_LR, aes(x=pred_model_backward)) +
  geom_density(lwd=0.5) +
  labs(title = "Distribution of Probability Prediction Data",x="Diabetes Probability",y="Density") +
  theme_minimal()
```



From the likelihood forecast graph above, it tends to be seen that most of the information are negative for diabetes.

Assessment

From testing execution utilizing testing dataset above, we can assess our model utilizing disarray lattice. Normally the mastrix utilized for model assessment is exactness, particularity, responsiveness, and accuracy. Yet, for this situation we center around awareness, or the correlation between the quantity of positive perceptions that are anticipated to be positive (True Positive) and the absolute number of perceptions that are really certain (True Positive + False Negative).

```
confusionMatrix_LR <- confusionMatrix(data = diabetes_test_LR$label_model_backward,reference = diabetes
```

```
## Warning in confusionMatrix.default(data =
## diabetes_test_LR$label_model_backward, : Levels are not in the same order for
## reference and data. Refactoring data to match.
```

```
confusionMatrix_LR
```

```
## Confusion Matrix and Statistics
##
##                Reference
## Prediction     Not Diabetes Diabetes
##   Not Diabetes           88       23
##   Diabetes               10       33
##
##                Accuracy : 0.7857
```

```
##                  95% CI : (0.7124, 0.8477)
##     No Information Rate : 0.6364
##     P-Value [Acc > NIR] : 4.573e-05
##
##                   Kappa : 0.5128
##
##  Mcnemar's Test P-Value : 0.03671
##
##             Sensitivity : 0.5893
##             Specificity : 0.8980
##          Pos Pred Value : 0.7674
##          Neg Pred Value : 0.7928
##              Prevalence : 0.3636
##          Detection Rate : 0.2143
##    Detection Prevalence : 0.2792
##       Balanced Accuracy : 0.7436
##
##        'Positive' Class : Diabetes
##
```

assumption test

Suspicions are basically conditions that ought to be met before we draw surmisings with respect to the model assessments. Coordinated factors relapse model requirements multicolinearity test to demonstrate that the subsequent model isn't misdirecting, or has one-sided assessors. Multicollinearity happens when the autonomous factors are excessively exceptionally related with one another.

Multicollinearity will be tried with Variance Inflation Factor (VIF). Change expansion element of the direct relapse is characterized as VIF = 1/Tolerance (T). With VIF $>$ 10 there is multicollinearity among the factors.

```
vif(LR_diabetes_model_backward)
```

```
##              Pregnancies                  Glucose            BloodPressure
##                 1.053118                 1.025641                 1.129475
##                      BMI DiabetesPedigreeFunction
##                 1.101066                 1.017861
```

From result above, we can see that each variabel has no connection since every one of our indicators that utilized for causing model to have VIF $<$ 10.

2.K-Nearest Neighbors

The rule of K-Nearest Neighbor (KNN) is to track down the nearest distance between the information to be assessed and the k nearest neighbors in the preparation information. Where k is the quantity of nearest neighbors.

Pre-Processing

In deciding the nearest neighbors, it is important to guarantee that the size of each mathematical indicator has something very similar or almost a similar scale. Assuming the size of every variable has a very different scale, it is important to level the size of every variable with the goal that the it are adjusted and fair to result results.

```
summary(diabetes)
```

```
##   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
##  Not Diabetes:500
##  Diabetes    :268
##
##
##
##
```

From the data above, it just so happens, the size of every variable has high contrast, so it is important to do scaling in view of the typical worth and standard deviation of every variable from the train dataset.

Predictors

```
diabetes_train_x <- diabetes_train %>% select(-Outcome)
diabetes_test_x <- diabetes_test %>% select(-Outcome)
```

Target

```
diabetes_train_y <- diabetes_train %>% select(Outcome)
diabetes_test_y <- diabetes_test %>% select(Outcome)
```

So the train and test dataset is a similar even in the wake of scaling, the scaling system should be finished utilizing similar boundaries. This implies scaling the train dataset and test to utilize the typical boundary and standard deviation of the train dataset for every indicator.

```
diabetes_train_x_scaled <- scale(x = diabetes_train_x)
diabetes_test_x_scaled <- scale(x = diabetes_test_x,
                                center = attr(diabetes_train_x_scaled,"scaled:center"),
                                scale = attr(diabetes_train_x_scaled,"scaled:scale"))
```

Then, at that point, to decide the quantity of k, for the most part it tends to be founded on the foundation of the quantity of train datasets.

```
k <- sqrt(nrow(diabetes_train))
k
```

```
## [1] 24.77902
```

```
k~25
```

```
## k ~ 25
```

predicting

Order strategies utilizing k-NN don't construct a model. All the more in fact, it can say that no 'boundaries' are found out about the information. Moreover, after every one of the indicators are scaled, and the k worth is gotten. Then, at that point, forecasts are made involving the boundaries as beneath.

```
diabetes_test_KNN <- diabetes_test
diabetes_train_KNN<-diabetes_train
glimpse(diabetes_test_KNN$label_predicted_KNN <- knn(train = diabetes_train_x_scaled,test =diabetes_tes
```

```
##  Factor w/ 2 levels "Not Diabetes",..: 1 2 1 1 1 1 1 2 1 1 ...
```

Assessment

The expectation results above are then contrasted and the test dataset for assessment of the outcomes. Also, similar to the Logistic Regression over, the framework to zero in on is responsiveness.

```
confusionMatrix_KNN <- confusionMatrix(data = diabetes_test_KNN$label_predicted_KNN,
                reference = diabetes_test_KNN$Outcome,positive = "Diabetes")
confusionMatrix_KNN
```

```
## Confusion Matrix and Statistics
##
##               Reference
## Prediction     Not Diabetes Diabetes
##   Not Diabetes           88       28
##   Diabetes               10       28
##
##                Accuracy : 0.7532
##                  95% CI : (0.6774, 0.8191)
##     No Information Rate : 0.6364
##     P-Value [Acc > NIR] : 0.001317
##
##                   Kappa : 0.4274
##
##  Mcnemar's Test P-Value : 0.005820
##
##             Sensitivity : 0.5000
##             Specificity : 0.8980
##          Pos Pred Value : 0.7368
##          Neg Pred Value : 0.7586
##              Prevalence : 0.3636
##          Detection Rate : 0.1818
##    Detection Prevalence : 0.2468
##       Balanced Accuracy : 0.6990
##
##        'Positive' Class : Diabetes
##
```

conclusion

From the consequences of the assessment of the two models above, it tends to be seen the responsiveness an incentive for every strategy underneath.

```
comparison_sensitivity <- data.frame("Sensitivity Logistic Regression"=data.frame(confusionMatrix_LR$by
           "Sensitivity K-Nearest Neighbors"=data.frame(confusionMatrix_KNN$byClass)[1,])
comparison_sensitivity
```

```
##   Sensitivity.Logistic.Regression Sensitivity.K.Nearest.Neighbors
## 1                       0.5892857                             0.5
```

From the correlation results above, apparently the aftereffects of the Logistic Regression have a superior awareness esteem (0.5892) than the consequences of KNN (0.5). This implies that KNN has POOR execution in grouping contrasted with Logistic Regression. KNN has a shortcoming since it is unimaginable to expect to know which boundaries/indicators have major areas of strength for an in foreseeing targets.

In this way, assuming we additionally focus on knowing the extent of indicators' effect on the model, utilizing Logistic Regression is better. However, assuming that we center more around the forecast results without focusing on the extent of indicators' impact on the model, then utilizing KNN is great.

Reference

1. 13. Engelgau M.M., Narayan K., Herman W.H. Screening for type 2 diabetes. Diabetes Care. 2000;23:1563–1580. doi: 10.2337/diacare.23.10.1563.

2. Ryden L., Standl E., Bartnik M., Van den Berghe G., Betteridge J., De Boer M.J., Cosentino F., Jönsson B., Laakso M., Malmberg K., et al. Guidelines on diabetes, pre-diabetes, and cardiovascular diseases: Executive summary: The Task Force on Diabetes and Cardiovascular Diseases of the European Society of Cardiology (ESC) and of the European Association for the Study of Diabetes (EASD) Eur. Heart J. 2007;28:88–136

3. Heikes K.E., Eddy D.M., Arondekar B., Schlessinger L. Diabetes Risk Calculator: A simple tool for detecting undiagnosed diabetes and pre-diabetes. Diabetes Care. 2008;31:1040–1045. doi: 10.2337/dc07-1150

4. Razavian N., Blecker S., Schmidt A.M., Smith-McLallen A., Nigam S., Sontag D. Population-level prediction of type 2 diabetes from claims data and analysis of risk factors. Big Data. 2015;3:277–287. doi: 10.1089/big.2015.0020

5. Knowler W.C., Barrett-Connor E., Fowler S.E., Hamman R.F., Lachin J.M., Walker E.A., Nathan D.M., Diabetes Prevention Program Research Group Reduction in the incidence of type 2 diabetes with lifestyle intervention or metformin. N. Engl. J. Med. 2002;346:393–403.