Predictive Modeling of Diabetes Occurrence in Females

Final Group Report

Project group 15: Tianyu Duan (Niki), Marcela Flaherty, Angela (Junyu) Chen, Rishavpreet Singh

Total word count: (1820 words)

Introduction

Diabetes, characterized by elevated blood glucose levels, stems from inadequate insulin production or inefficient utilization. Type 1 diabetes involves immune system attacks on insulin-producing cells, necessitating lifelong insulin injections, while Type 2, prevalent in adults, is linked to insulin resistance or inadequate production, often tied to lifestyle factors. Symptoms encompass increased thirst, frequent urination, fatigue, and blurred vision. Gestational diabetes during pregnancy heightens Type 2 diabetes risk postpartum. Managing diabetes involves medication, lifestyle changes, and regular monitoring. In women, diabetes is influenced by factors like hormonal shifts during pregnancy, advancing age, high BMI, genetic predisposition, and hypertension, highlighting the need for tailored healthcare.

This study explores the multifaceted factors affecting diabetes in women with the goal of developing a nuanced understanding for informed preventive measures and interventions. Specifically, it aims to examine the association between the response variable (Diabetes) and explanatory variables (pregnancies, insulin, glucose, blood pressure, BMI, diabetes pedigree function, and age) among females, using predictive modeling. The primary focus is on building a model to forecast diabetes occurrence in females based on these variables, addressing the central question: **How can predictive modeling be employed to forecast the occurrence of diabetes in females, exploring the association between the response variable (Diabetes) and explanatory variables such as pregnancies, insulin, glucose, blood pressure, BMI, diabetes pedigree function, and age?**

Dataset description:

The dataset used is picked from Kaggle (https://www.kaggle.com/datasets/whenamancodes/predict-diabities?resource=download) and contains 9 variables and 768 instances with the information on blood pressure, diabetes status, and other health-related variables of females. It is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. Based on the diagnostic measurements included in the dataset, its objective is to predict whether a patient has diabetes. The dataset has been selected from a larger database using several constraints. Particularly, all patients in the dataset are females, 21 years of age or older, belonging to the Pima Indian heritage.

Dataset attributes:

Columns	Description
Pregnancies	To express the Number of pregnancies
Glucose	To express the Glucose level in blood
BloodPressure	To express the Blood pressure measurement
SkinThickness	To express the thickness of the skin
Insulin	To express the Insulin level in blood
BMI	To express the Body mass index
DiabetesPedigreeFunction	To express the Diabetes percentage
Age	To express the age
Outcome	To express the final result 1 is Yes and 0 is No

Methods and Results

a) Exploratory Data Analysis (EDA)

In [1]:

```
# Load all the required libraries.
library(tidyverse)
library(repr)
library(ggplot2)
library(cowplot)
library(moderndive)
library(broom)
library(GGally)
library(digest)
library(infer)
library(gridExtra)
library(caret)
library(pROC)
library(boot)
library(glmnet)
library(tidymodels)
# Setting a seed to ensure reproducibility
set.seed(1)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.
0 ──
✔ dplyr     1.1.3     ✔ readr     2.1.4
✔ forcats   1.0.0     ✔ stringr   1.5.0
✔ ggplot2   3.4.3     ✔ tibble    3.2.1
✔ lubridate 1.9.3     ✔ tidyr     1.3.0
✔ purrr     1.0.2     
── Conflicts ────────────────────────────────────────── tidyverse_conflicts
() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all c
onflicts to become errors

Attaching package: ‘cowplot’


The following object is masked from ‘package:lubridate’:

    stamp


Registered S3 method overwritten by 'GGally':
  method from   
  +.gg   ggplot2


Attaching package: ‘gridExtra’


The following object is masked from ‘package:dplyr’:

    combine


Loading required package: lattice


Attaching package: ‘caret’


The following object is masked from ‘package:purrr’:

    lift


Type 'citation("pROC")' for a citation.


Attaching package: ‘pROC’


The following objects are masked from ‘package:stats’:

    cov, smooth, var
```

```
Attaching package: ‘boot’


The following object is masked from ‘package:lattice’:

    melanoma


Loading required package: Matrix


Attaching package: ‘Matrix’


The following objects are masked from ‘package:tidyr’:

    expand, pack, unpack


Loaded glmnet 4.1-8

── Attaching packages ────────────────────────────────────── tidymodels 1.1.
1 ──

✔ dials        1.2.0     ✔ tune         1.1.2
✔ modeldata    1.2.0     ✔ workflows    1.1.3
✔ parsnip      1.1.1     ✔ workflowsets 1.0.1
✔ recipes      1.0.8     ✔ yardstick    1.2.0
✔ rsample      1.2.0

── Conflicts ───────────────────────────────────────── tidymodels_conflicts
() ──
✖ gridExtra::combine()     masks dplyr::combine()
✖ scales::discard()        masks purrr::discard()
✖ Matrix::expand()         masks tidyr::expand()
✖ dplyr::filter()          masks stats::filter()
✖ recipes::fixed()         masks stringr::fixed()
✖ dplyr::lag()             masks stats::lag()
✖ caret::lift()            masks purrr::lift()
✖ Matrix::pack()           masks tidyr::pack()
✖ yardstick::precision()   masks caret::precision()
✖ yardstick::recall()      masks caret::recall()
✖ yardstick::sensitivity() masks caret::sensitivity()
✖ yardstick::spec()        masks readr::spec()
✖ yardstick::specificity() masks caret::specificity()
✖ recipes::step()          masks stats::step()
✖ Matrix::unpack()         masks tidyr::unpack()
✖ recipes::update()        masks Matrix::update(), stats::update()
• Search for functions across packages at https://www.tidymodels.org/find/
```

In [2]:

```
# Read the data from the web into R
diabetes_data <- read_csv("https://raw.githubusercontent.com/plotly/datasets
# Viewing the data
head(diabetes_data)
```

A tibble: 6 × 9

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFu
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	
6	148	72	35	0	33.6	
1	85	66	29	0	26.6	
8	183	64	0	0	23.3	
1	89	66	23	94	28.1	
0	137	40	35	168	43.1	
5	116	74	0	0	25.6	

In [3]:

```
# Basic information of the data
nrow(diabetes_data)
# Check missing values
colSums(diabetes_data==0)
```

768

Pregnancies: 111 **Glucose:** 5 **BloodPressure:** 35 **SkinThickness:** 227 **Insulin:** 374 **BMI:** 11 **DiabetesPedigreeFunction:** 0 **Age:** 0 **Outcome:** 500

In this case, the variables Glucose, BloodPressure, SkinThickness, Insulin, and BMI with values of 0 are deemed unreasonable and considered as missing values. Therefore, we will eliminate rows with 0 values in Glucose, BloodPressure, and BMI. However, there are 227 missing values for SkinThickness and 374 missing values for Insulin; removing too many rows would result in a very small dataset. There is a trade-off between keeping more variables and retaining more examples. Excluding the variables Insulin and SkinThickness may lead to a less robust model, but including these variables will result in a smaller dataset for training and testing the model. After evaluating different models among group members, we decided to keep the Insulin variable and remove the SkinThickness variable in this final report.

In [4]:

```
# Filter out BloodPressure = 0, Insulin = 0, Glucose = 0 and BMI = 0
diabetes_data_filtered <- diabetes_data %>%
filter(BloodPressure != 0, Glucose != 0, BMI != 0, Insulin != 0)

# Total number of examples in the filtered data
nrow(diabetes_data_filtered)
```

392

In [5]:

```
# Select the columns we want to use and rename the column name Outcome to Di
diabetes_data_selected <- diabetes_data_filtered %>%
    select(Pregnancies, Glucose, BloodPressure, BMI, DiabetesPedigreeFunctic
    rename(Diabetes = Outcome)
```

```
# Create another dataset for the visualization
diabetes_data_final <- diabetes_data_selected %>% mutate(Diabetes = ifelse(c
diabetes_data_final$Diabetes <- as.factor(diabetes_data_final$Diabetes)
head(diabetes_data_final)
```

A tibble: 6 × 8

Pregnancies	Glucose	BloodPressure	BMI	DiabetesPedigreeFunction	Age	Insulin
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	89	66	28.1	0.167	21	94
0	137	40	43.1	2.288	33	168
3	78	50	31.0	0.248	26	88
2	197	70	30.5	0.158	53	543
1	189	60	30.1	0.398	59	846
5	166	72	25.8	0.587	51	175

Data Visualization & Summary Table:

In [6]:

```
# Create a pairplot to show the relationships between the explanatory variab
options(repr.plot.width = 10, repr.plot.height = 10)
diabetes_data_final_pairplots <- diabetes_data_final %>%
  ggpairs(progress = FALSE, aes(colour=Diabetes), lower=list(combo=wrap("fac
  theme(
    text = element_text(size = 10),
    plot.title = element_text(face = "bold"),
    axis.title = element_text(face = "bold")
  )
diabetes_data_final_pairplots
```

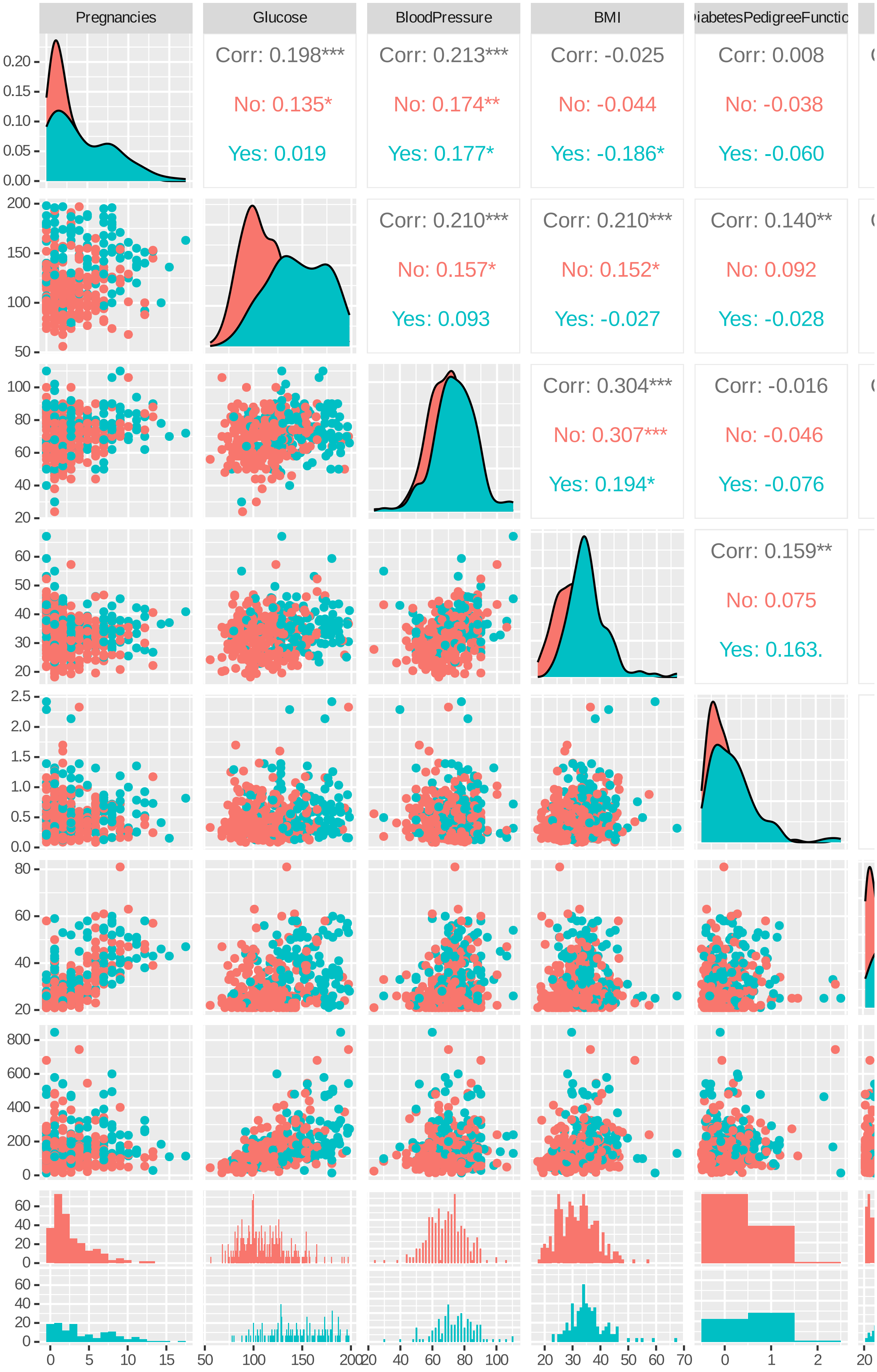
Pregnancies
Glucose
BloodPressure
BMI
iabetesPedigreeFunctio
Corr: 0.198***
No: 0.135*
Yes: 0.019
Corr: 0.213***
No: 0.174**
Yes: 0.177*
Corr: -0.025
No: -0.044
Yes: -0.186*
Corr: 0.008
No: -0.038
Yes: -0.060
Corr: 0.210***
No: 0.157*
Yes: 0.093
Corr: 0.210***
No: 0.152*
Yes: -0.027
Corr: 0.140**
No: 0.092
Yes: -0.028
Corr: 0.304***
No: 0.307***
Yes: 0.194*
Corr: -0.016
No: -0.046
Yes: -0.076
Corr: 0.159**
No: 0.075
Yes: 0.163.
0.20
0.15
0.10
0.05
0.00
200
150
100
50
100
80
60
40
20
60
50
40
30
20
2.5
2.0
1.5
1.0
0.5
0.0
80
60
40
20
800
600
400
200
0
60
40
20
0
60
40
20
0
0
5
10
15
50
100
150
200
20
40
60
80
100
20
30
40
50
60
70
0
1
2
20

Figure 1. Pairplot for Exploring Relationships Among Explanatory Variables for Diabetic and Non-Diabetic Females

```
In [7]: # Compute the means of different explanatory variables for Diabetic and Non-
        estimates <- diabetes_data_final %>%
            group_by(Diabetes) %>%
            summarise_at(vars(Pregnancies, Glucose, BloodPressure, Insulin, BMI, Dia

        estimates
```

```
                                                 A tibble: 2 × 8
Diabetes  Pregnancies_mean  Glucose_mean  BloodPressure_mean  Insulin_mean  BMI_r
   <fct>             <dbl>         <dbl>               <dbl>         <dbl>      <
      No          2.721374      111.4313            68.96947      130.8550   31.7
     Yes          4.469231      145.1923            74.07692      206.8462   35.7
```

Table 1. Mean Values of Explanatory Variables for Diabetic and Non-Diabetic Females

Based on the plot above (Figure 1), no explanatory variables showed a high correlation (greater than 0.6 in absolute value) with each other, which means that there is no potential problem of multicollinearity. This observation is crucial for our predictive modeling, as multicollinearity can lead to unstable and unreliable estimates of variable relationships. Therefore, we can be confident that the relationships between our explanatory variables are not confounded by high correlations.

In addition, when looking at the boxplots and the summary table above (Table 1), the mean values of Pregnancies, Insulin, Glucose, BloodPressure, BMI, DiabetesPedigreeFunction, and Age are all higher in females with diabetes compared to those without diabetes. This suggests that these variables may play a significant role in distinguishing between the two groups. This information highlights potential key predictors of diabetes in females.

Moreover, the histograms and line graphs provide further insights into the distributions of these six variables. They show that the distributions of these variables are more skewed to the right for non-diabetic females compared to those with diabetes. This skewness indicates potential differences in the distribution of these variables among the two groups. For instance, the skewness might imply that certain variables, like Pregnancies or Age, have a more pronounced impact on diabetes risk.

In our further analysis, we will explore these variables in greater depth, investigate potential outliers, and test their significance in predicting diabetes in females. By examining these variables and their distributions, we aim to gain a deeper understanding of the factors contributing to diabetes in females and refine our predictive model.

b) Methods: Plan

Logistic regression is a suitable method for predicting the likelihood of diabetes occurrence in females, as it explores the association between the response variable (Diabetes) and explanatory variables such as pregnancies, insulin, glucose, blood pressure, BMI, diabetes pedigree function, and age. Its efficacy lies in its compatibility with binary outcome variables, making it adept at predicting the probability of events, such as the presence or absence of diabetes.

Before implementing any method, the data will be divided into two parts: the training set and the test set. This division allows for the building and refinement of the regression model using the training data. In this analysis, three different types of logistic regression models will be explored: (1) ordinary logistic regression, (2) ridge logistic regression, and (3) LASSO regression.

In [8]:

```
# Split data into training (0.7) and test sets (0.3)
Diabetes_split <- initial_split(diabetes_data_selected, prop = 0.7, strata =
training_diabetes <- training(Diabetes_split)
testing_diabetes <- testing(Diabetes_split)
nrow(training_diabetes)
nrow(testing_diabetes)
```

```
274
118
```

In [9]:

```
# Prepare the model matrix for glmnet
model_matrix_X_train <-
    model.matrix(object = Diabetes ~ ., data = training_diabetes)[ , -1]

matrix_Y_train <-
    as.matrix(training_diabetes$Diabetes, ncol = 1)
```

(1) Ordinary Logistic Regression

Ordinary logistic regression, being the standard model without additional regularization, estimates coefficients for each predictor variable.

In [10]:

```
# Fit a logistic regression model using the glm function
diabetes_logistic_model <- glm(
    formula = Diabetes ~.,
    data = training_diabetes,
    family = binomial)

tidy_result <- tidy(diabetes_logistic_model)
tidy_result$p.value <- round(tidy_result$p.value, 3)
tidy_result
```

A tibble: 8 × 5

term	estimate	std.error	statistic	p.value
<chr>	<dbl>	<dbl>	<dbl>	<dbl>
(Intercept)	-8.999698e+00	1.327328813	-6.780308057	0.000
Pregnancies	9.354026e-02	0.063415461	1.475038798	0.140
Glucose	3.047096e-02	0.006525991	4.669169834	0.000
BloodPressure	3.942689e-03	0.014221915	0.277226280	0.782
BMI	6.536550e-02	0.025112780	2.602878105	0.009
DiabetesPedigreeFunction	8.118797e-01	0.459205630	1.768008936	0.077
Age	3.628650e-02	0.021564440	1.682700731	0.092
Insulin	-1.039082e-05	0.001446740	-0.007182229	0.994

Table 2. Logistic Regression Coefficients and P-values Summary

Referencing Table 2, for ordinary logistic regression model, the p-values for Pregnancies, BloodPressure, DiabetesPedigreeFunction, Age, and Insulin exceed 0.05. This suggests that these predictor variables may not be statistically significant in predicting the response variable (Diabetes). Nevertheless, it is crucial to recognize that statistical significance does not necessarily imply practical significance, and vice versa. Even if a variable lacks statistical significance, its effect size (coefficient value) and the context of the problem should be considered for a practical interpretation.

(2) Ridge Logistic Regression

Ridge logistic regression, incorporating L2 regularization, adds a penalty term to coefficients to prevent them from becoming too large, addressing multicollinearity. Ridge regression is suitable when avoiding overfitting while maintaining all predictors in the model is essential.

In [11]:

```
# Ridge Regression
Diabetes_cv_lambda_ridge <-
  cv.glmnet(
       x = model_matrix_X_train,
       y = matrix_Y_train,
       alpha = 0,
       family = 'binomial',
       type.measure = 'auc',
       nfolds = 10)

Diabetes_cv_lambda_ridge
```

```
Call:  cv.glmnet(x = model_matrix_X_train, y = matrix_Y_train, type.measure
= "auc",      nfolds = 10, alpha = 0, family = "binomial")

Measure: AUC

    Lambda Index Measure      SE Nonzero
min   0.04    95  0.8447 0.03215       7
1se 200.69     2  0.8407 0.03346       7
```

Next, we will obtain the $\hat{\lambda}_{\text{min}}$ which provides the maximum average AUC out of the whole sequence for λ.

In [12]:

```
# Obtain λ min using Diabetes_cv_lambda_ridge:
Diabetes_lambda_max_AUC_ridge <- round(Diabetes_cv_lambda_ridge$lambda.min,

# Fit the ridge regression model the optimum value for λ
Diabetes_ridge_max_AUC <-
  glmnet(
  x = model_matrix_X_train, y = matrix_Y_train,
  alpha = 0,
  family = 'binomial',
  lambda = Diabetes_lambda_max_AUC_ridge
)
```

(3) LASSO Regression

LASSO logistic regression, with L1 regularization, includes a penalty term based on the absolute values of coefficients. It possesses a feature selection property, driving some coefficients exactly to zero and eliminating irrelevant variables. LASSO is particularly useful when suspecting that many predictors may not significantly contribute to the outcome, and a more interpretable model is desired.

In [13]:

```
# LASSO Regression
Diabetes_cv_lambda_LASSO <-
  cv.glmnet(
  x = model_matrix_X_train, y = matrix_Y_train,
  alpha = 1,
  family = 'binomial',
  type.measure = 'auc',
  nfolds = 5)
Diabetes_cv_lambda_LASSO
```

```
Call:  cv.glmnet(x = model_matrix_X_train, y = matrix_Y_train, type.measure
= "auc",      nfolds = 5, alpha = 1, family = "binomial")

Measure: AUC

     Lambda Index Measure      SE Nonzero
min 0.00773    37  0.8266 0.04784       6
1se 0.15181     5  0.7798 0.04792       2
```

In [14]:

```
# Obtain λ 1se using Diabetes_cv_lambda_LASSO:
Diabetes_lambda_max_AUC_LASSO <- round(Diabetes_cv_lambda_LASSO$lambda.min,
# Fit the LASSO Regression model the optimum value for λ
Diabetes_LASSO_max_AUC <- glmnet(
  x = model_matrix_X_train, y = matrix_Y_train,
  alpha = 1,
  family = 'binomial',
  lambda = Diabetes_lambda_max_AUC_LASSO
)
coef(Diabetes_LASSO_max_AUC)
```

```
8 x 1 sparse Matrix of class "dgCMatrix"
                                  s0
(Intercept)              -8.235984689
Pregnancies               0.078989691
Glucose                   0.028652698
BloodPressure             0.001994353
BMI                       0.058437531
DiabetesPedigreeFunction  0.657044093
Age                       0.035783566
Insulin                   .
```

The coefficient for Insulin shown above is marked as '.', indicating that its estimated coefficient is exactly zero. In LASSO regression, this suggests that this variable is excluded from the model and does not contribute to predicting the response variable.

In [15]:

```
# Compute the Cross-validation AUC for the ordinary logistic model
num.folds <- 10

folds <- createFolds(training_diabetes$Diabetes, k=num.folds)

regr.cv <- NULL
for (fold in 1:num.folds) {
train.idx <- setdiff(1:nrow(training_diabetes), folds[[fold]])
regr.cv[[fold]] <- glm(Diabetes ~ ., data=training_diabetes, subset=train.id
                       family="binomial")
    }

pred.cv <- NULL
auc.cv <- numeric(num.folds)

for (fold in 1:num.folds) {
test.idx <- folds[[fold]]
pred.cv[[fold]] <- data.frame(obs=training_diabetes$Diabetes[test.idx],
pred=predict(regr.cv[[fold]], newdata=training_diabetes, type="response")[te
auc.cv[fold] <- roc(obs ~ pred, data=pred.cv[[fold]])$auc
    }

Diabetes_cv_ordinary <- round(mean(auc.cv),7)
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases

Setting levels: control = 0, case = 1

Setting direction: controls < cases
```

In [16]:

```
# Create a table for Cross-validation AUC of different logistic models
Diabetes_AUC_models <-
    tibble(
        model = c("ordinary", "ridge", "lasso"),
        auc = c(Diabetes_cv_ordinary,
                max(Diabetes_cv_lambda_ridge$cvm), max(Diabetes_cv_lambda_LA
Diabetes_AUC_models
```

A tibble: 3 × 2

model	auc
<chr>	<dbl>
ordinary	0.8232978
ridge	0.8446836
lasso	0.8266359

Table 3. AUC Scores from Cross-Validation for Each Model

Based on the CV results presented in Table 3, the ridge regression model exhibits the highest cross-validation AUC, followed by the LASSO regression, and lastly, the ordinary regression model. Therefore, we will opt for the ridge regression model, anticipating superior prediction performance.

Next, we will use the ridge regression model to predict the target variable (Diabetes) on the test set (testing_diabetes). Subsequently, plot the ROC curve based on the test set, including the AUC (AUC measures the classification ability of the classifier, a higher the AUC means a better predictive performance) to evaluate the model's overall performance.

In [17]:

```r
model_matrix_X_test <-
    model.matrix(object = Diabetes ~ .,
                 data = testing_diabetes)[, -1]

ROC_ridge <-
    roc(
        response = testing_diabetes$Diabetes,
        predictor = predict(Diabetes_ridge_max_AUC,
                    newx = model_matrix_X_test)[,"s0"] )
ROC_ridge

options(repr.plot.width = 8, repr.plot.height = 8)

plot(ROC_ridge,
  print.auc = TRUE, col = "blue", lwd = 3, lty = 2,
  main = "ROC Curve of Ridge model in the test set for Diabetes Dataset"
)
```

```
Setting levels: control = 0, case = 1

Setting direction: controls < cases
```

```
Call:
roc.default(response = testing_diabetes$Diabetes, predictor = predict(Diabet
es_ridge_max_AUC,     newx = model_matrix_X_test)[, "s0"])

Data: predict(Diabetes_ridge_max_AUC, newx = model_matrix_X_test)[, "s0"] in
79 controls (testing_diabetes$Diabetes 0) < 39 cases (testing_diabetes$Diabe
tes 1).
Area under the curve: 0.8916
```

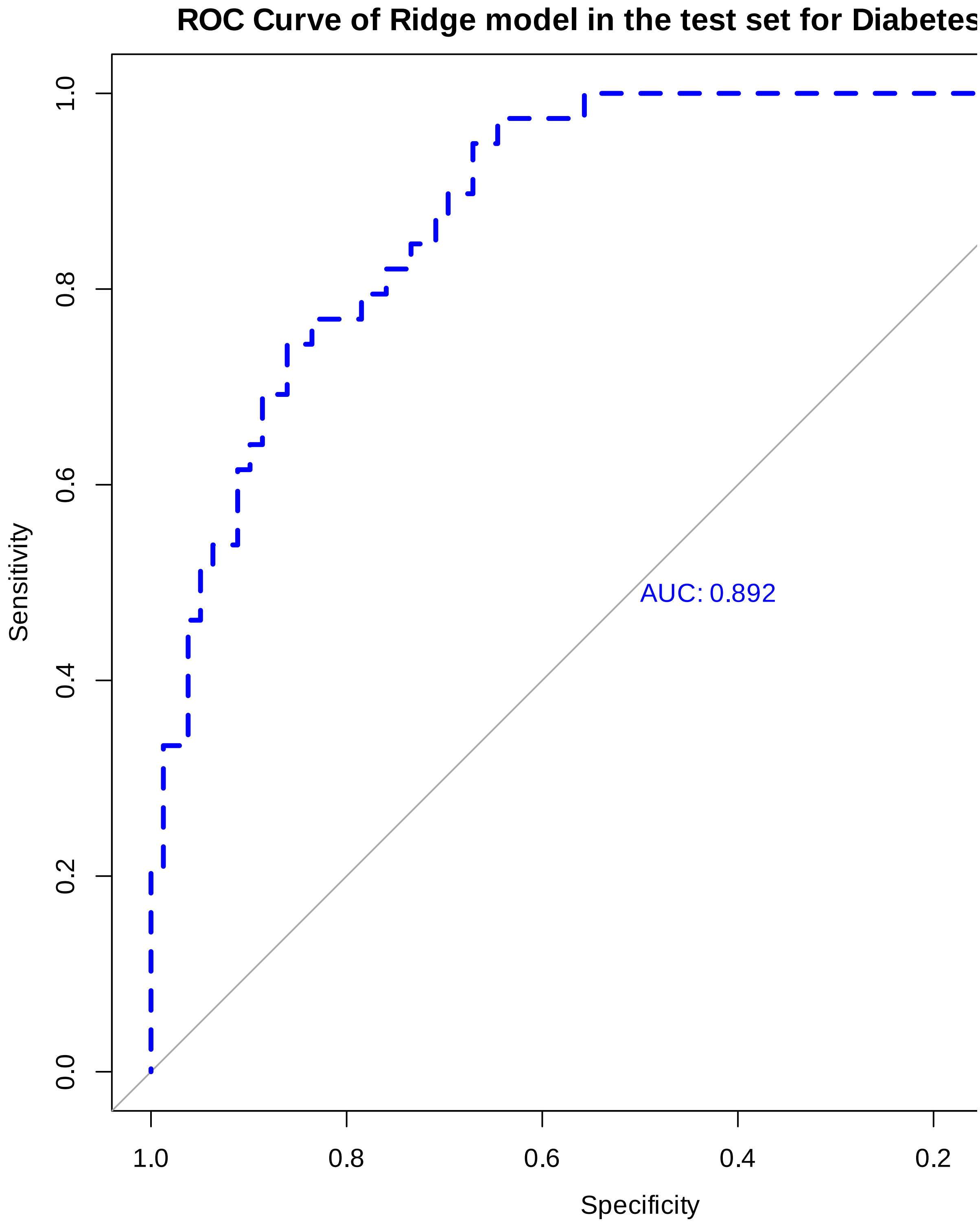

Figure 2. ROC Curve of Ridge model in the test set for Diabetes Dataset

Interpretation of the results:

The Ridge Regression model demonstrated robust performance on the test data, achieving an AUC of 0.892. This notable AUC score implies that the predictive model, incorporating crucial variables such as pregnancies, insulin, glucose, blood pressure, BMI, diabetes pedigree function, and age, exhibits strong discriminatory capabilities. It effectively distinguishes between instances of diabetes and non-diabetes among females. The heightened AUC underscores the model's accuracy in correctly classifying individuals into their respective groups. The inclusion of diverse explanatory variables emphasizes the model's comprehensive understanding of the intricate relationships within the dataset, enhancing its predictive efficacy. While these results highlight the potential utility of the Ridge Regression model in accurately identifying diabetic cases among the studied population, it's essential to note the limited dataset size—only 274 examples were used for training, with 118 observations for testing.

Discussion

In our analysis, we employed three logistic regression methods to predict diabetes occurrence in Pima Indian females over the age of 21. Alongside our dependent variable Diabetes, we focused on independent variables Pregnancies, Glucose, BloodPressure, BMI, DiabetesPedigreeFunction, Age, and Insulin. Our dataset, reduced to 392 instances due to the deletion of rows with zero values except for the Pregnancies variable, presented both opportunities and challenges. Beginning with a 70/30 split into training and testing data, we then implemented an early analysis of ordinary, lasso, and ridge logistic regression. The ordinary logistic regression model suggested limited statistical significance for several predictors. Ridge regression, which addresses multicollinearity through penalty coefficients, showed the highest AUC score, indicating its effectiveness in reducing overfitting. LASSO regression, recognized for its feature selection property, identified the independent variable Insulin as non-contributory. Through a cross-validation AUC of the different logistic models, we discovered that the ridge regression model has the highest cross-validation AUC (0.8446836), followed by the LASSO regression (0.8266359), with the minimum value being of the ordinary regression model (0.8232978). Subsequently, we began an in-depth exploration of our ridge regression model by fitting our model onto our prediction with our testing dataset. By plotting our prediction on a ROC curve against our AUC, it was clear that our chosen ridge regression model has a strong performance (AUC 0.892), supporting our expected findings.

Our findings have potential limitations, such as the reduced dataset size. We found this data filtration step to be important for data quality and removing skew or bias, as missing values had been replaced with zeros in this real-world dataset. Although beneficial, this step significantly reduced our sample size, potentially impacting our model's power and generalizability. Increasing the dataset size would enhance the robustness of our model. In addition to data size, several assumptions underlie logistic regression, including the independence of observations, linearity of independent variables and log odds, and the absence of multicollinearity. These assumptions are crucial to ensuring the reliability and validity of the model's predictions. Given the logistic regression's limitations, future studies could explore alternative predictive models, regularization techniques, or feature selection methods. These models might better capture the variable interactions in diabetes prediction.

Alongside model improvements, there are future explorations in which our study could lead to answering a variety of research questions with real-world applications. Investigating how diabetes risk factors change over time in individuals could offer valuable insights for modelling. Integrating data to include different demographics in addition to Pima Indian females over 21 would allow for more generalizable insights, with application to larger populations. Further investigation into additional variables such as different health markers or lifestyle choices could reveal deeper insights into diabetes development and its predictors. The integration of these approaches would allow for more complex predictive modelling of diabetes and contribute to the broader field of predictive health analytics, answering potential research questions on diabetes factors concerning a wide range of ethnic, gender, and age backgrounds.

References

- Chauhan, A. (2022). Predict Diabetes. Kaggle. https://www.kaggle.com/datasets/whenamancodes/predict-diabities?resource=download
- Cleveland Clinic. (n.d.). Diabetes. https://my.clevelandclinic.org/health/diseases/7104-diabetes
- Centers for Disease Control and Prevention. (2022, June 20). Diabetes and women. https://www.cdc.gov/diabetes/library/features/diabetes-and-women.html
- Gotter, A. (2022, October 13). How diabetes affects women. Healthline. https://www.healthline.com/health/diabetes/symptoms-in-women#pregnancy
- Ratner, R. E., Christophi, C. A., Metzger, B. E., Dabelea, D., Bennett, P. H., Pi-Sunyer, X., Fowler, S., Kahn, S. E., & The Diabetes Prevention Program Research Group. (2008). Prevention of Diabetes in Women with a History of Gestational Diabetes: Effects of Metformin and Lifestyle Interventions. *The Journal of Clinical Endocrinology & Metabolism, 93(12)*, 4774–4779. https://doi.org/10.1210/jc.2008-0772
- Berbudi, A., Rahmadika, N., Tjahjadi, A. I., & Ruslami, R. (2020). Type 2 diabetes and its impact on the immune system. *Current Diabetes Reviews, 16(5)*, 442–449. https://doi.org/10.2174/1573399815666191024085838