A Logistic Regression Equation to Predict the Onset of Diabetes in Pima Indian Women

Girish Palya (425998) and Karina Norvoish ()

2020-06-05

Objective

To develop and validate an empirical equation in order to predict the onset of diabetes in Pima Indian women, and compare the results to a prior attempt by Smith et al.[1]

[1] JW Smith, JE Everhart, WC Dicksont, WC Knowler, RS Johannes. 1988. *Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus*

Research Design and Methods

A predictive equation was developed using logistic regression analysis and data collected from 768 Pima Indian women. The equation incorporated age, BMI, capillary plasma glucose, a hereditary function, and pregnancy count as independent covariates for diagnosing onset of diabetes within 5 years. The equation was evaluated using binned residual plots[2]. Its predictive performance was compared against the results obtained by Smith et al.

[2] A. Gelman and J. Hill, *Data Analysis Using Regression and Multilevel/Hierarchical Models*, Cambridge University Press, 2007.

Results

The predictive equation was calculated using logistic regression parameters as follows: $P(\textit{diagnosis} = 1) = 1/(1 - e^{-x})$, where $x =$ -19.457 + 0.046(*blood glucose in mg/dl*) + 3.352(log(*BMI in kg/in*)) + 0.118(*pregnancy count*) + 4.175(*hereditary factor*) + 0.012(*age in years*) - 0.025(*blood glucose : hereditary factor*). At a threshold of 0.5 for positive diagnosis, equation's sensitivity was 88%, specificity was 59% and error rate was 22%. At 0.35 cut-off point for positive diagnosis, equation's sensitivity was 76% and specificity was 74%, while error rate remains the same. When the model was trained using 576 randomly selected cases (coefficients of logistic regression recalculated) and prediction was performed on the remaining 192 cases, sensitivity remains the same while specificity varies in the range of 72-74% (after 100 repetitions). The area under ROC curve was 84%. In contrast, Smith et al. report sensitivity and specificity of 76% when ADAP algorithm was trained on 576 cases. They did not report error rate or the area under ROC curve; only the shape of the ROC curve was reported.

Conclusion

Performance of logistic regression model compares favorably with neural network models like ADAP learning algorithm. Careful selection of input variables, transformations, and interactions can result in a logistic regression model whose performance will be on par with more sophisticated techniques.

DIABETES MELLITUS afflicts nearly 9% of world population (463 million) and causes 4 million deaths every year. The ability to forecast is central to many medical situations involving care and management. Although many sophisticated models have been developed for discriminant analysis, recent empirical comparisons indicate that standard methods such as logistic regression work very well[3].

[3] C B Begg. *Statistical Methods in Medical Diagnosis*

Data and Related Work

SMITH ET AL.[4] have used the ADAP Learning Algorithm to forecast the onset of diabetes mellitus. They describe ADAP as "an adaptive learning routine that generates and executes digital analogs of perceptron-like devices".

[4] JW Smith, JE Everhart, WC Dickson, WC Knowler, RS Johannes. 1988. *Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus*

Data used by Smith et al. is available from Kaggle via UCI Machine Learning Repository.

A subset of data.

```
diabetes <- read.csv(
  str_c("https://raw.githubusercontent.com/",
        "girishji/Pima/master/data/diabetes.csv"))
head(diabetes) %>% kable()
```

Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DPF	Age	Diagnosis
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0

Study Population

THE POPULATION for this study was the Pima Indian population near Phoenix, Arizona. That population has been under continuous study since 1965 by the National Institute of Diabetes and Digestive and Kidney Diseases because of its high incidence rate of diabetes. Each community resident over 5 years of age was asked to undergo a standardized examination every two years, which included an oral glucose tolerance test. Diabetes was diagnosed according to World Health Organization Criteria[5]; that is, if the 2 hour post-load plasma glucose was at least 200 mg/dl (11.1 mmol/l) at any survey examination or if the Indian Health Service Hospital serving the community found a glucose concentration of at least 200 mg/dl during the course of routine medical care[6].

[5] World Health Organization, *Report of a Study Group: Diabetes Mellitus*. World Health Organization Technical Report Series. Geneva, 727, 1985.

[6] Knowler, W.C., P.H. Bennett, RF. Hamman, and M. Milier. 1978. *Diabetes incidence and prevalence in Pima Indians: a 19-fold greater incidence than in Rochester*, Minnesota. Am J Epidemiol 108:497-505.

Variables

THE SUBJECTS of the study are Pima Indian women over 21 years of age. The following explanatory variables are found to be risk factors for diabetes.

1. `Pregnancies`: Number of times pregnant
2. `Glucose`: Plasma Glucose Concentration at 2 Hours in an Oral Glucose Tolerance Test (GTIT) (in mg/dl)
3. `BloodPressure`: Diastolic Blood Pressure (mm Hg)
4. `SkinThickness`: Triceps Skin Fold Thickness (mm)
5. `Insulin`: 2-Hour Serum Insulin (Uh/ml)
6. `BMI`: Body Mass Index (Weight in kg / (Height in in))
7. `DPF`[7]: Diabetes Pedigree Function in the range of (0, 1)
8. `Age`: Age (years)

[7] Diabetes Pedigree Function was developed by Smith et al.* to provide a measure of the expected genetic influence of affected and unaffected relatives on the subject's eventual diabetes risk. It uses information from parents, grandparents, full and half siblings, full and half aunts and uncles, and first cousins.

* JW Smith, JE Everhart, WC Dicksont, WC Knowler, RS Johannes. 1988. *Using the ADAP Learning Algorithm to Forecast the Onset of Diabetes Mellitus*

The binary dependent variable (`Diagnosis`) describes the onset of non-insulin-dependent diabetes mellitus (DM) within a five-year period. This variable is 1 if diagnosis is positive within five years, and 0 otherwise.

Observations from Data

PRELIMINARY EXAMINATION of data reveals the following:

- There are many spurious 0 values in the data, especially in `Insulin` and `SkinThickness`. These values skew the mean and cause excessive outliers. Zero values are presumed to be errors in data, and suitable remedies will be employed to address the skew.
- There are 500 negative instances (65.1%) of the regressand (`Diagnosis`), compared to 258 positive instances (34.9%). Even though only logit

model is considered for this study, using probit model will likely produce a similar result since the CDFs underlying binomial logit and probit models differ most in the tails.

- Density plot indicate that not all distributions are normal. Data is skewed towards younger subjects. `Pregnancies` are also skewed towards subjects having fewer pregnancies. `Insulin` and `SkinThickness` are distored by spurious zero values. `BMI`, `Insulin`, and `SkinThickness` have long tails on the right side.

Zero values in some variables indicate errors in data.

```
diabetes %>%
  summarise_each(~ sum(.x == 0)) %>%
  pivot_longer(everything(), names_to = 'Variable',
               values_to = 'Count of Zero Values in Column') %>%
  knitr::kable()
```

Variable	Count of Zero Values in Column
Pregnancies	111
Glucose	5
BloodPressure	35
SkinThickness	227
Insulin	374
BMI	11
DPF	0
Age	0
Diagnosis	500

```
diabetes %>%
  select_at(vars(!Diagnosis)) %>%
  pivot_longer(everything(), names_to = 'Variable',
               values_to = 'Value') %>%
  ggplot(aes(x = Variable, y = Value,
             fill = Variable)) +
  geom_jitter(size = 0.1, alpha = 0.2) +
  stat_summary(fun = mean, geom = "point",
               shape = 20, size = 4,
               color = "red", fill = "red") +
  geom_boxplot(alpha = 0.3) +
  theme_light() +
  theme(legend.position="none") +
  xlab('') +
  ylab('')
```

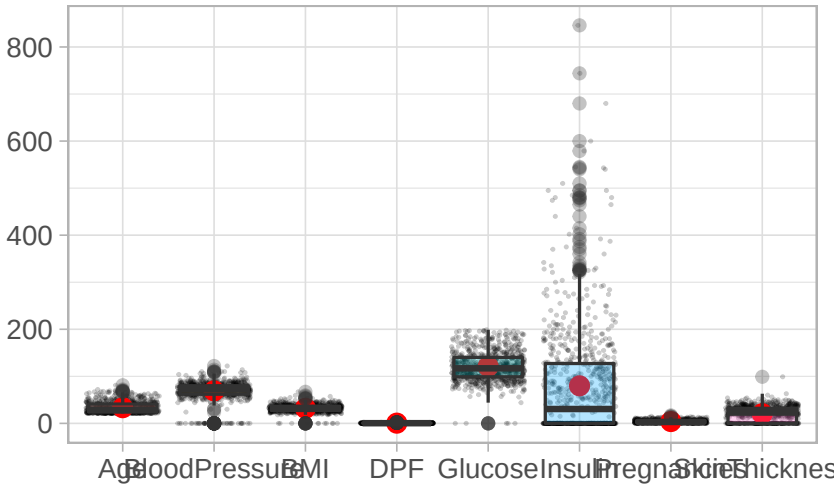


Figure 1: Box plot showing spurious outliers in 'Insulin' and 'SkinThickness', and skewed mean (red dot).

```
diabetes %>%
  select_at(vars(!Diagnosis)) %>%
  pivot_longer(everything(), names_to='Regressor',
               values_to='Value') %>%
  ggplot(aes(x = Value, group = Regressor,
             fill = Regressor)) +
  geom_density(alpha = 0.3) +
  facet_wrap(~Regressor, scales = "free", nrow = 2) +
  theme_light() +
  theme(legend.position="none") +
  xlab('') +
  ylab('')

# Correlation matrix displayed on the right side.
diabetes %>%
  rename(Preg = 1, BloodPr = 3, SkinTh = 4) %>%
  GGally::ggcorr(method = c("everything", "pearson"))
```

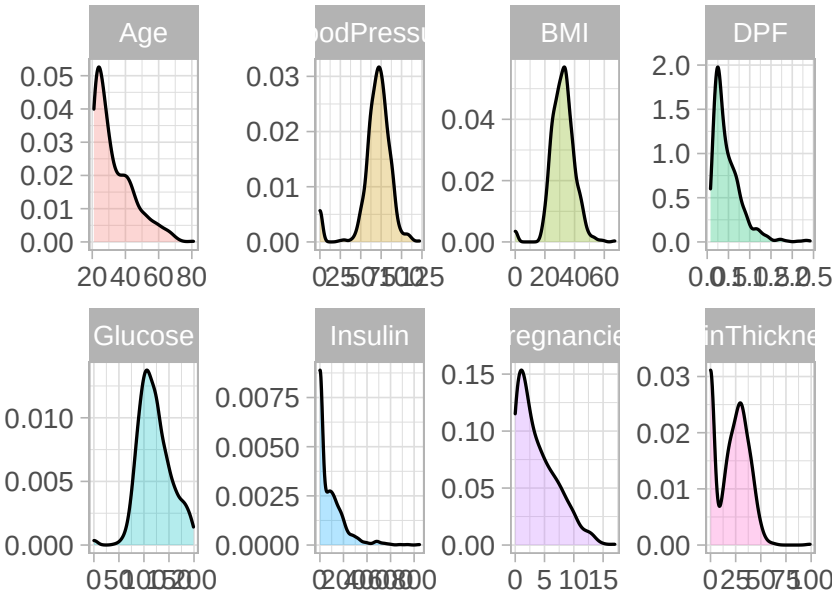


Figure 2: Density plot of regressors (including spurious zero values).

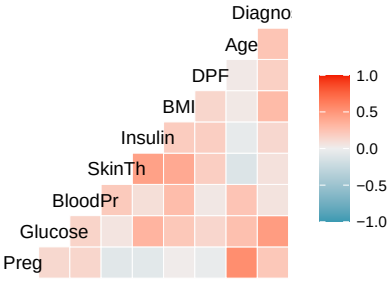


Figure 3: Correlation among regressors.

Scatter plots and correlation matrix reveal the following:

- Among all the regressors, `Glucose` concentration has the highest correlation (.467) with onset of diabetes.
- `Age` is not strongly correlated (.238) with onset of diabetes. This seems to suggest that, either accumulation of unhealthy lifestyle habits is not a factor, or that such factors have already expressed themselves by the age of 21 years.
- Diabetes Pedigree Function (`DPF`) is not highly correlated with onset of diabetes, which suggests that hereditary factors may be less important.
- `Insulin` and `SkinThickness` show significant correlation (.437), but this may be a result of both variables having too many spurious zero values.
- Presence of spurious zero values in `Insulin` could be masking its correlation with `Glucose`.
- None of the explanatory variables are correlated with the response variable in a dominating way.

```
diabetes %>%
  GGally::ggpairs(., lower = list(continuous =
```

```
    GGally::wrap("points",
        alpha = 0.3, size = 0.1)))
```

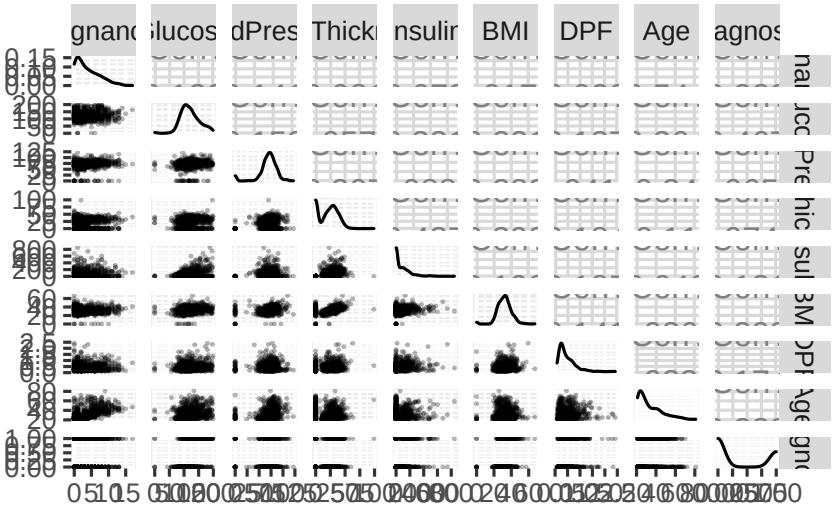


Figure 4: Scatter plots and correlation matrix of regressors.

Model

Logistic regression is the standard way to model binary outcomes. The probability that $Diagnosis = 1$ is modeled as

$$P(Diagnosis_i = 1) = logit^{-1}(X_i),$$

under the assumption that the outcomes $Diagnosis_i$ are independent given these probabilities. X is the linear predictor. The function $logit^{-1}(x) = \frac{e^x}{1+e^x}$ transforms the continuous values to the range (0, 1), which is necessary since probabilities must be between 0 and 1.

In building the regression model all input variables that are expected to play a significant role in predicting the outcome are included; Their interactions are also included if the variables have large effects. Statistically non-significant variables are included as long as they have expected sign in the coefficients, and excluded if they do not have expected sign.

The strategy outlined by Gelman and Hill[8] is followed since it is intuitive. Essentially, a simple model is considered first and additional complexity is progressively included, taking care to check for problems along the way.

[8] page 69, Andrew Gelman and Jennifer Hill, *Data Analysis Using Regression and Multilevel/Hierarchical Models, 2006.*

Logistic regression with just one predictor

GLUCOSE LEVEL IN BLOOD is known to be a strong predictor of onset of diabetes, and the correlation matrix reflects this relationship.

Fitting the model using just `Glucose`:[9]

[9] display() is a modification of arm::display() to remove the echo of formula.

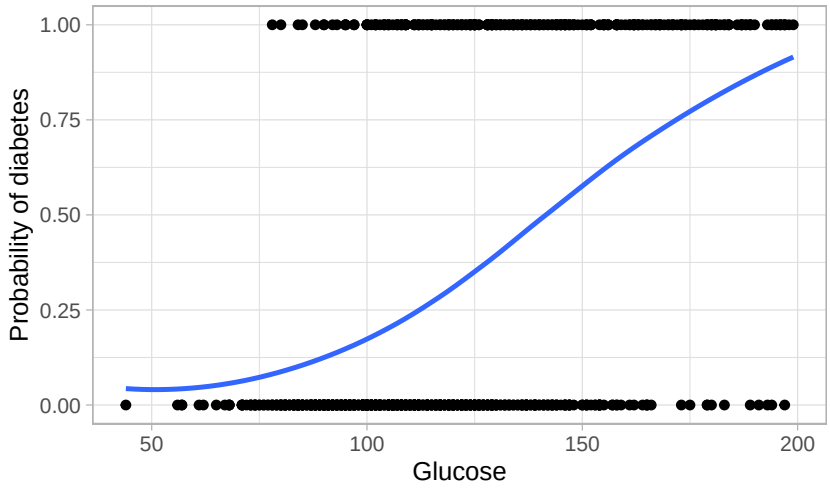
```
fit.1 <- glm(Diagnosis ~ Glucose, diabetes,
             family = binomial(link = "logit"))
display(fit.1)
```

```
##             coef.est coef.se
## (Intercept) -5.3501   0.4208
## Glucose      0.0379   0.0033
## ---
##   n = 768, k = 2
##   residual deviance = 808.7, null deviance = 993.5
##   (difference = 184.8)
```

The coefficient for `Glucose` is .0379, which seems low, but it is measured in mg/dl. The mean value for `Glucose` is 120 mg/dl. Multiplicative effect is still significant.

Graphing the fitted model

```
diabetes %>% mutate(Predicted = fitted(fit.1)) %>%
  filter(Glucose != 0) %>%
  ggplot(aes(Glucose, Diagnosis)) + geom_point() +
  stat_smooth(aes(Glucose, Predicted), se = F) +
  ylab('Probability of diabetes') + theme_light()
```




```
# Best-fit and uncertainity of logistic regression
#   shown on the right side.
sim.1 <- arm::sim(fit.1)
plot_uncertainity <- function(.data) {
```

```
  plt <- .data %>%
    mutate(Predicted = fitted(fit.1)) %>%
    filter(Glucose != 0)
  for (j in 1:10) {
    plt <- plt %>%
      mutate(!!sym(str_c('s_', j)) :=
          arm::invlogit(sim.1@coef[j,1] +
                        sim.1@coef[j,2] * Glucose))
  }
  plt <- plt %>%
    ggplot(aes(Glucose, Diagnosis)) + geom_point()
  for (j in 1:10) {
    plt <- plt + stat_smooth(aes(Glucose,
                    !!sym(str_c('s_', j))),
                        se = F, color = "gray")
  }
  plt <- plt +
    stat_smooth(aes(Glucose, Predicted), se = F) +
    labs(y = "Probability (diabetes)", title = "") +
    theme_light()
  return(plt)
}
plot_uncertainity(diabetes)
```

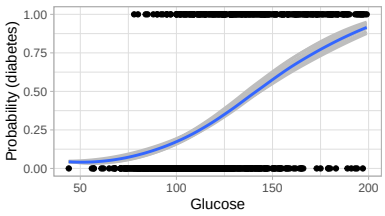


Figure 5: In the range of data, the solid line shows the best-fit logistic regression, and the light lines show uncertainty in the fit.

Interpreting the logistic regression coefficients

Our model is

$$P(diabetes) = logit^{-1}(-5.3501 + 0.0379 \cdot Glucose)$$

- The constant term can be interpreted when `Glucose` $= 0$, in which case the probability of diagnosing diabetes is $logit^{-1}(-5.3501) = 0.0047$. However `Glucose` level at 0 does not make sense, so constant term is not interpreted.
- Predictive difference with respect to `Glucose` is evaluated by computing the derivative at the average value of `Glucose` in the data set, which is 120.9 mg/dl. The value of the linear predictor here is -5.3501 + 0.0379 $\cdot$ 120.9 = -0.76799, and so the slope of the curve at this point is $0.0379e^{-0.76799}/(1 + e^{-0.76799})^2 = 0.0082$. Thus, adding 10 mg/dl to `Glucose` corresponds to a positive difference in the probability of diagnosing diabetes by about 8.2%.
- Considering the statistical significance of the coefficient for `Glucose`, the slope is estimated well; the standard error is only 0.0033, which

is tiny compared to the coefficient estimate of 0.0379. The approximate 95% (2 standard errors) interval is [0.0313, 0.0445]. This is statistically significant.

Adding a second input variable

We extend the model by adding BMI. The coefficient is expected to be positive.

```
fit.2 <- glm(Diagnosis ~ Glucose + BMI, diabetes,
             family = binomial(link = "logit"))
display(fit.2)
```

```
##             coef.est coef.se
## (Intercept) -7.5156   0.6052
## Glucose      0.0352   0.0033
## BMI          0.0763   0.0133
## ---
##   n = 768, k = 3
##   residual deviance = 771.4, null deviance = 993.5
##   (difference = 222.1)
```

Comparing two individuals, the one with 1 mg/dl higher blood glucose will encounter 0.0352 logit probability of diabetes diagnosis. Similarly, an increase of 1 kg/in in BMI corresponds to an increase of 0.0763 logit probability of diagnosis. Both coefficients are statistically significant, each being more than 2 standard errors away from zero. And both their signs make sense: glucose level and BMI are known risk factors for diabetes.

For a quick interpretation, the "divide by 4 rule"[10] comes handy. Applying the rule on the coefficients, 1 mg/dl increase in glucose leads to increas in probability of diabetes diagnosis by 0.88%. Similarly, 1 kg/in increase in BMI increases probability of diabetes diagnosis by 1.9%.

Comparing these two coefficients, it may appear that `BMI` is more a important factor. But this is **incorrect**. The standard deviation of `BMI` is 7.9 and for `Glucose` it is 31.9. The logistic regression coefficients corresponding to 1-standard-deviation differences are 0.0352x31.9 for `Glucose` and 0.0763x7.9 for `BMI` respectively. Again, applying the "divide by 4 rule", 1-standard deviation difference of `Glucose` yields a 28% increase in probability, while in `BMI` it is only 15%.

[10] page 82, Andrew Gelman and Jennifer Hill, *Data Analysis Using Regression and Multilevel/Hierarchical Models, 2006.*

Comparing the coefficient estimates when adding a predictor

The coefficient for `Glucose` changes from 0.379 in the original model to 0.352. This is because people who have high `Glucose` are likely to have higher `BMI`. The two factors have a small positive correlation (0.221) as indicated in the correlation matrix.

Adding additional input variables

PREGNANCIES is added to the model and checked for significance.

```
fit.3 <- glm(Diagnosis ~ Glucose + BMI + Pregnancies,
             diabetes, family = binomial(link = "logit"))
display(fit.3)
```

```
##             coef.est coef.se
## (Intercept) -8.1240   0.6385
## Glucose      0.0342   0.0033
## BMI          0.0816   0.0137
## Pregnancies  0.1371   0.0268
## ---
##   n = 768, k = 4
##   residual deviance = 744.1, null deviance = 993.5
##   (difference = 249.4)
```

Number of pregnancies has small standard error and is statistically significant from 0 at 95% interval (0.0835, 0.1907). It is not obvious why high number of pregnancies, in and of itself, is a risk factor.

DPF, the hereditary factor, is added to the model. Although diabetes mellitus is not genetic per se, DNA may influence the risk of developing it. This type of diabetes tends to run in families.

```
fit.4 <- glm(Diagnosis ~ Glucose + BMI + Pregnancies + DPF,
             diabetes, family = binomial(link = "logit"))
display(fit.4)
```

```
##             coef.est coef.se
## (Intercept) -8.4159   0.6569
## Glucose      0.0338   0.0033
## BMI          0.0781   0.0138
## Pregnancies  0.1419   0.0271
## DPF          0.9013   0.2917
## ---
##   n = 768, k = 5
##   residual deviance = 734.3, null deviance = 993.5
##   (difference = 259.2)
```

DPF has a positive coefficient as expected, but has some standard error. At 95% interval of (0.3179, 1.4847) it is still statistically significant from 0.

Age is generally not known to be a risk factor for this type of diabetes.

```
fit.5 <- glm(Diagnosis ~ Glucose + BMI + Pregnancies +
               DPF + Age,
             diabetes, family = binomial(link = "logit"))
display(fit.5)
```

```
##             coef.est coef.se
## (Intercept) -8.6731   0.6900
## Glucose      0.0329   0.0034
## BMI          0.0795   0.0138
## Pregnancies  0.1195   0.0317
## DPF          0.8915   0.2922
## Age          0.0122   0.0091
## ---
##   n = 768, k = 6
##   residual deviance = 732.5, null deviance = 993.5
##   (difference = 261.0)
```

`Age` is not statistically significant, since it has a large standard error. However, it has the correct sign in the coefficient. This input may not influence the predictive power of the model very much but will not hurt either. It will not be discarded.

Blood Pressure is added to the model and checked for significance.

```
fit.5 <- glm(Diagnosis ~ Glucose + BMI + Pregnancies +
               DPF + Age + BloodPressure,
             diabetes, family = binomial(link = "logit"))
display(fit.5)
```

```
##               coef.est coef.se
## (Intercept)   -8.2398   0.7020
## Glucose        0.0335   0.0034
## BMI            0.0877   0.0143
## Pregnancies    0.1249   0.0320
## DPF            0.8962   0.2949
## Age            0.0163   0.0092
## BloodPressure -0.0135   0.0051
## ---
```

```
##   n = 768, k = 7
##   residual deviance = 725.5, null deviance = 993.5
##   (difference = 268.0)
```

`BloodPressure` is only marginally statistically significant from 0. The standard error is large, and consequently, its 95% interval is [-0.0237, -0.0033]. Moreover, it has -ve coefficient. It is known that diabetes damages arteries and makes them targets for hardening, called atherosclerosis. This can cause high blood pressure, which would indicate a +ve sign for the coefficient. This is a dubious input, and we will removed from the model.

SkinThickness and Insulin are added to the model. It is worth noting beforehand that both of these inputs have proportionately large number of zeros in their columns.

```
fit.6 <- glm(Diagnosis ~ Glucose + BMI + Pregnancies +
               DPF + Age + SkinThickness + Insulin,
             diabetes, family = binomial(link = "logit"))
display(fit.6)
```

```
##               coef.est coef.se
## (Intercept)   -8.8369   0.7046
## Glucose        0.0345   0.0037
## BMI            0.0840   0.0148
## Pregnancies    0.1178   0.0317
## DPF            0.9566   0.2970
## Age            0.0105   0.0091
## SkinThickness -0.0025   0.0067
## Insulin       -0.0011   0.0009
## ---
##   n = 768, k = 8
##   residual deviance = 730.0, null deviance = 993.5
##   (difference = 263.5)
```

Both `SkinThickness` and `Insulin` suffer from large standard error and are not statistically significant from 0.

`SkinThickness` measures subcutaneous fat; sign of the coefficient is expected to be positive. Given the unexpected -ve sign of the coefficient and large standard error, in addition to the occurrence of zeros in the data, this input variable is discarded from the model.

`Insulin` is also excluded from the model since it has large standard error and large number of observation errors.

Our model so far is as follows:

```
fit.7 <- glm(Diagnosis ~ Glucose + BMI + Pregnancies +
               DPF + Age,
```

```
            diabetes, family = binomial(link = "logit"))
stargazer::stargazer(fit.7, type = 'text', single.row = TRUE,
                     dep.var.caption = "")
```

```
##
## ================================================
##                              Diagnosis
## ------------------------------------------------
## Glucose                  0.033*** (0.003)
## BMI                      0.080*** (0.014)
## Pregnancies              0.119*** (0.032)
## DPF                      0.891*** (0.292)
## Age                        0.012 (0.009)
## Constant                 -8.673*** (0.690)
## ------------------------------------------------
## Observations                    768
## Log Likelihood                -366.254
## Akaike Inf. Crit.              744.509
## ================================================
## Note:              *p<0.1; **p<0.05; ***p<0.01
```

Check for interactions

A QUICK SEARCH OF ALL POSSIBLE TWO-FACTOR INTERACTIONS based on reducing AIC values reveals the following candidates for inclusion.

It is not clear if any of these interactions will improve prediction, or that they even make sense in combination. More insight is needed about their behavior. This topic will be revisited later.

```
search = step(fit.7, ~.^2)
```

```
search$anova
```

```
##                Step Df  Deviance Resid. Df Resid. Dev      AIC
## 1                    NA        NA       762   732.5088 744.5088
## 2 + Pregnancies:Age -1 10.613469       761   721.8953 735.8953
## 3     + Glucose:DPF -1  7.017733       760   714.8776 730.8776
## 4     + Glucose:Age -1  4.257862       759   710.6197 728.6197
## 5 + BMI:Pregnancies -1  2.761888       758   707.8579 727.8579
```

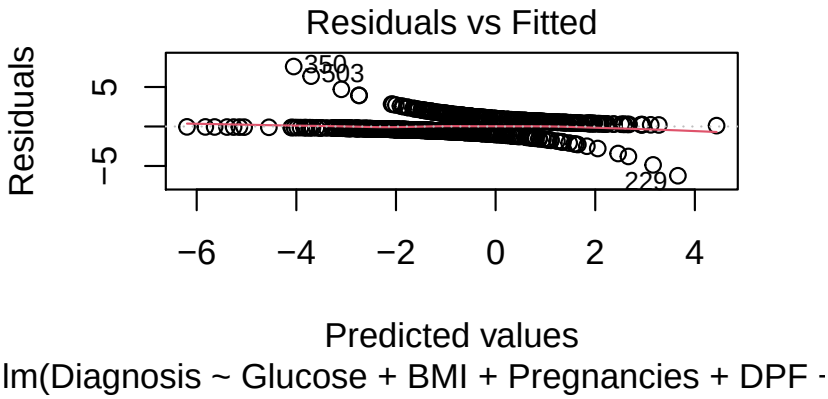
Evaluating, checking, and comparing fitted logistic regressions

RESIDUALS for logistic regression are defined as observed minus expected values:

$$residual_i = y_i - E(y_i|X_i) = y_i - logit^{-1}(X_i).$$

Data values y_i are discrete, and $logit^{-1}(X_i)$ values are continuous. The value of residuals are also discrete and depends on whether y_i is 0 or 1. Therefore, plots of raw residuals from logistic regressions are not useful.

```
plot(fit.7, which = 1)
```



A binned residual plot[11] is better.
From the authors:

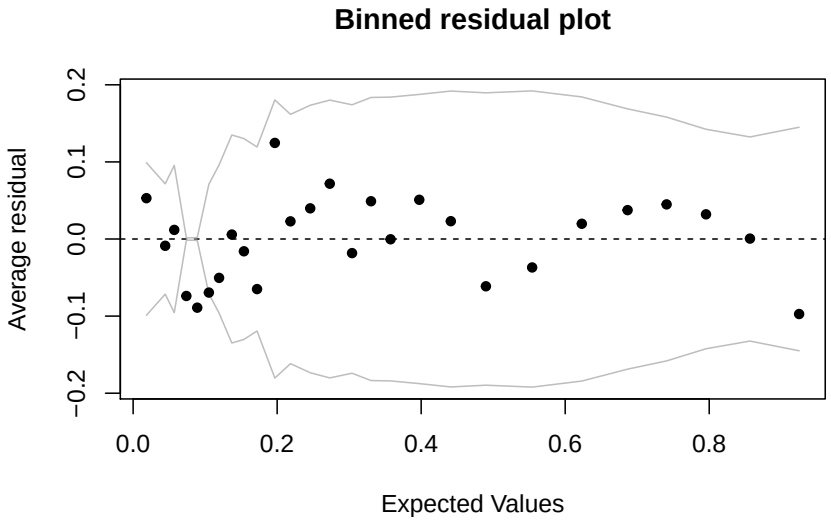
> We plot binned residuals by dividing the data into categories (bins) based on their fitted values, and then plotting the average residual versus the average fitted value for each bin. … here we divided the data into 40 bins of equal size. The dotted lines (computed as $2\sqrt{p(1-p)/n}$, where n is the number of points per bin) indicate ±2 standard-error bounds, within which one would expect about 95% of the binned residuals to fall, if the model were actually true.
>
> The bins are not equally spaced; rather, each bin has an equal number of data points. The light lines in the binned residual plot indicate theoretical 95% error bounds.

[11] page 97, Andrew Gelman and Jennifer Hill, *Data Analysis Using Regression and Multilevel/Hierarchical Models, 2006.*

In the model, 3 out of 27 bins fall outside 95% error bounds. This means roughly 3 out of every 27 predictions are incorrect. All the outliers are in the lower left quadrant. Residual is the difference between actual and predicted values. Negative residuals below 0.2 indicates that at low expected values our model overpredicts diabetes (more false positives). This is also observed later from ROC curve.

The dotted lines in the binned residual plot indicate theoretical 95% error bounds that would be appropriate if the model were true.

```
arm::binnedplot(fitted(fit.7),
                residuals(fit.7, type = "response"))
```



Plotting binned residuals versus inputs of interest

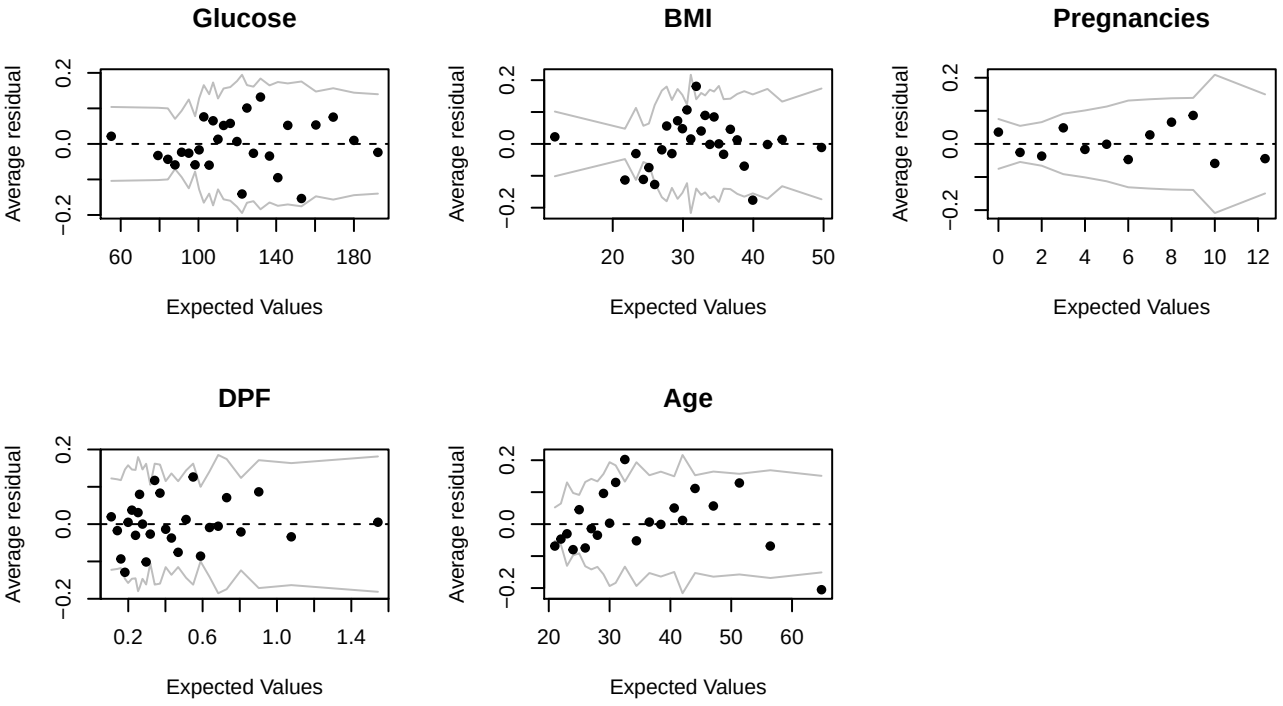
To understand the deviation better, residuals are binned and with respect to individual input variables, and plotted.

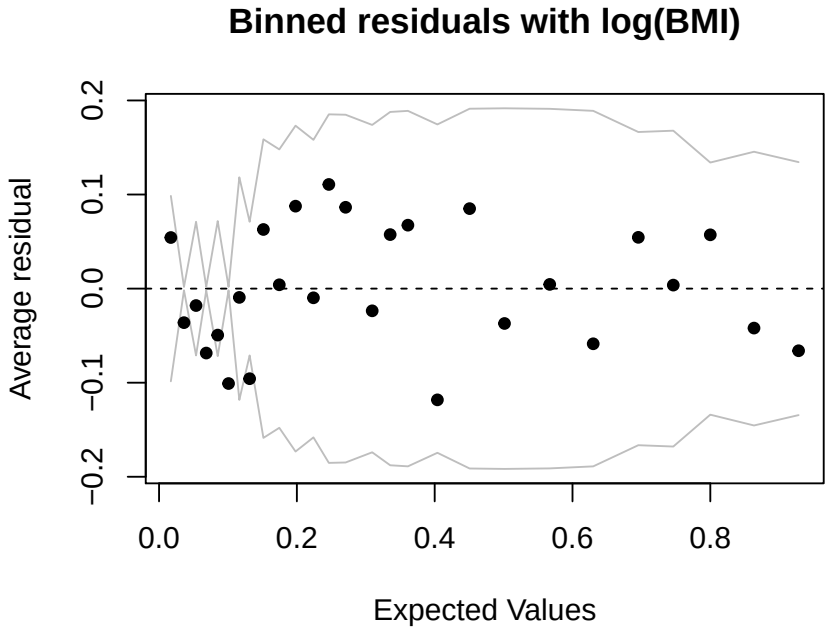
```
par(mfrow = c(2, 3))
for (inp in c('Glucose', 'BMI', 'Pregnancies', 'DPF', 'Age')) {
  arm::binnedplot(diabetes[[inp]],
                  residuals(fit.7, type = "response"),
                  main = inp)
}
```

Only `BMI` and `Age` exhibit outliers. `BMI` has 6 outliers and 4 of them are on the lower left quadrant (similar to the fitted model).

The raising and falling nature of residuals of `BMI` suggests that a log transformation on `BMI` may help. Applying log transformation (after temporarily removing 11 spurious 0 values in the data) reduces outlier count from 6 to 4.

```
fit.8 <- glm(Diagnosis ~ Glucose + log(BMI) + Pregnancies +
               DPF + Age,
             diabetes %>% filter(BMI != 0),
             family = binomial(link = "logit"))
arm::binnedplot(fitted(fit.8),
                residuals(fit.8, type = "response"),
                main = 'Binned residuals with log(BMI)')
```





To evaluate if log transformation improves predictive power, a comparison of confusion matrix is considered.

Confusion matrix, so named because the matrix summarizes how the model is confused, summarizes different types of model errors, such as false positives (Type 1 Error) and false negatives (Type 2 Error).

Applying log transformation makes false positives jump from 119 to 120, and false negatives reduce from 70 to 68. This is only a marginal improvement if any. However, it does not hurt to keep the log transformation on `BMI`.

```
# With log(BMI)
caret::confusionMatrix(
  factor(if_else(fitted(fit.8) > 0.34, 1, 0)),
  factor(diabetes %>% filter(BMI != 0) %>%
           pull(Diagnosis)))$table
```

```
##           Reference
## Prediction   0   1
##          0 371  68
##          1 120 198
```

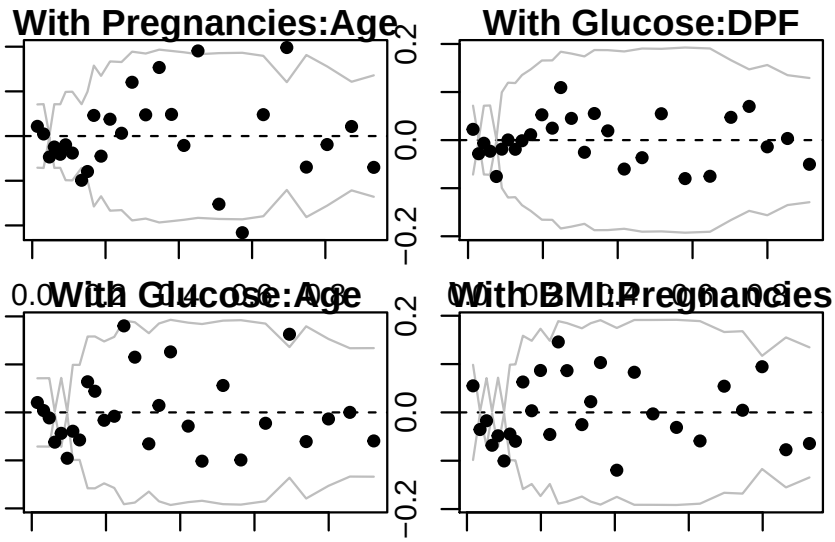
```
# Without any transformation on BMI
caret::confusionMatrix(
  factor(if_else(fitted(fit.7) > 0.34, 1, 0)),
  factor(diabetes %>% pull(Diagnosis)))$table
```

```
##           Reference
## Prediction   0   1
##          0 381  70
##          1 119 198
```

Check for interactions (again)

Combining inputs leads to increase or decrease in the influence produced by individual inputs, depending on the sign of the coefficient of the combined variable. From AIC analysis before we have 4 candidates for inclusion, namely, `Pregnancies:Age`, `Glucose:DPF`, `Glucose:Age`, and `BMI:Pregnancies`.

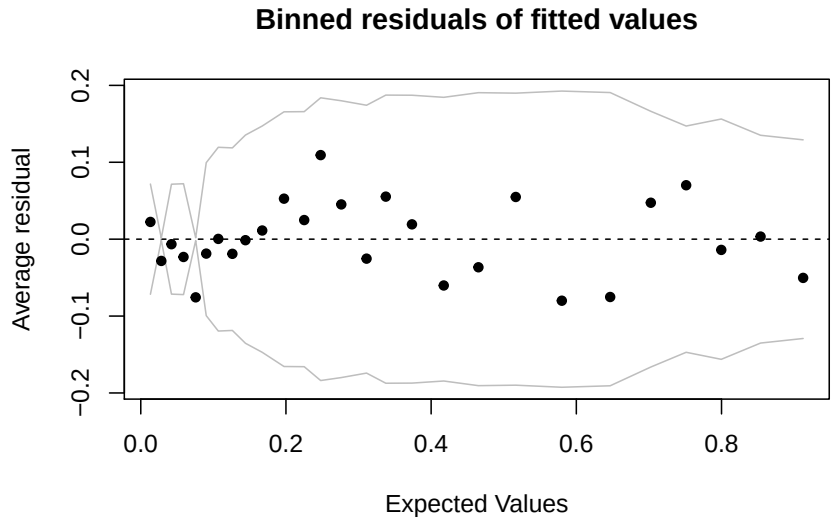
```
base <- "Diagnosis ~ Glucose + log(BMI) + Pregnancies + DPF + Age + "
par(mar = c(1, 1, 1, 1), mfrow = c(2, 2))
for (inter in c('Pregnancies:Age', 'Glucose:DPF',
                'Glucose:Age', 'BMI:Pregnancies')) {
  fit.temp <- glm(str_c(base, inter),
                  diabetes %>% filter(BMI != 0),
                  family = binomial(link = "logit"))
  arm::binnedplot(fitted(fit.temp),
                  residuals(fit.temp, type = "response"),
                  main = str_c('With ', inter))
}
```



`Glucose:DPF` is the most promising with only 2 outliers. Others have at least 3 outliers. `Glucose:DPF` also has tighter arrangement of residuals around 0 (horizontal axis). `DPF` influences `Glucose`, a highly significant input, in a meaningful way (discussed later).

Incorporating `Glucose:DPF` into the model, we observe that there are only 2 outliers close to 95% error boundary. This is a reasonably good model.

```
fit.9 <- glm(Diagnosis ~ Glucose + log(BMI) + Pregnancies +
               DPF + Age + Glucose:DPF,
             diabetes %>% filter(BMI != 0),
             family = binomial(link = "logit"))
arm::binnedplot(fitted(fit.9),
                residuals(fit.9, type = "response"),
                main = 'Binned residuals of fitted values')
```



Results

Description of the model

In summary, our model is as follows:

$$P(Diagnosis_i = 1) = logit^{-1}(X_i)$$

where,

$$X = -19.457 + 0.046(Glucose) + 3.352(\log(BMI)) + 0.118(Pregnancies) + 4.175(DPF) + 0.012(Age) - 0.025(Glucose : DPF)$$

```
formula <- Diagnosis ~ Glucose + log(BMI) + Pregnancies +
               DPF + Age + DPF:Glucose
fit.f <- glm(formula, diabetes %>% filter(BMI != 0),
             family = binomial(link = "logit"))
stargazer::stargazer(fit.f, type = 'text', single.row = TRUE,
                     dep.var.caption = "")
```

```
##
## ================================================
##                              Diagnosis
## ------------------------------------------------
## Glucose                  0.046*** (0.006)
## log(BMI)                 3.352*** (0.506)
## Pregnancies              0.118*** (0.032)
## DPF                      4.175*** (1.126)
## Age                        0.012 (0.009)
## Glucose:DPF              -0.025*** (0.008)
## Constant                -19.457*** (2.032)
## ------------------------------------------------
## Observations                    757
## Log Likelihood               -353.482
## Akaike Inf. Crit.             720.963
## ================================================
## Note:              *p<0.1; **p<0.05; ***p<0.01
```

```
fit.f$coefficients
```

```
##  (Intercept)      Glucose     log(BMI)  Pregnancies          DPF          Age
## -19.45683604   0.04614602   3.35224211   0.11755030   4.17489886   0.01213236
##  Glucose:DPF
##  -0.02506450
```

```
fit.f$coefficients[[3]]
```

```
## [1] 3.352242
```

The sign of the coefficient of `Glucose:DPF` is negative. This interaction can be interpreted in two ways:

- For every additional 10% (0.1) increase of hereditary proclivity to diabetes (`DPF`), a value of 0.0025 is subtracted from the coefficient of `Glucose`. Subtracting from the (positive) coefficient of `Glucose` leads to reduction in the effect on `Glucose` in predicting the onset of diabetes. Effect of blood glucose wane as hereditary effects become more prominent.
- For every additional 10 mg/dl increase in blood glucose, a value of 0.25 is subtracted from the coefficient of `DPF`. This is consistent with the observation that as blood glucose level increases (due to poor eating habits, for instance) hereditary plays less of a role in determining the onset of diabetes.

`BMI` has a multiplicative relationship with dependent variable `Diagnosis` owing to the log transformation. For every 10% increase `BMI`, linear predictor increases by 3.352xlog(1.1)=0.32. An inverse logit function will reveal the increase in shadow (dependent) variable.

Interpretation of other coefficients is already covered in previous sections.

Error rate and comparison to the null model

The error rate is defined as the proportion of cases for which the deterministic prediction $y_i = 1$ if $logit^{-1}(X_i) >= 0.5$ and guessing $y_i = 0$ if $logit^{-1}(X_i) < 0.5$ is wrong.

Our model has an error rate of 22% (compared to 35% for null model).

```
# Our model
error_rate <- function(predicted, reference, threshold = 0.5) {
  round(sum((predicted >= threshold & reference == 0) |
              (predicted < threshold & reference == 1)) /
          length(reference) * 100, digits = 2)
}
error_rate(fitted(fit.f), diabetes %>% filter(BMI != 0) %>%
             pull(Diagnosis))
```

```
## [1] 22.19
```

```
# Null model
round(min(sum(diabetes$Diagnosis == 1),
          sum(diabetes$Diagnosis == 0)) /
        length(diabetes$Diagnosis) * 100, digits = 2)
```

```
## [1] 34.9
```

ROC curve

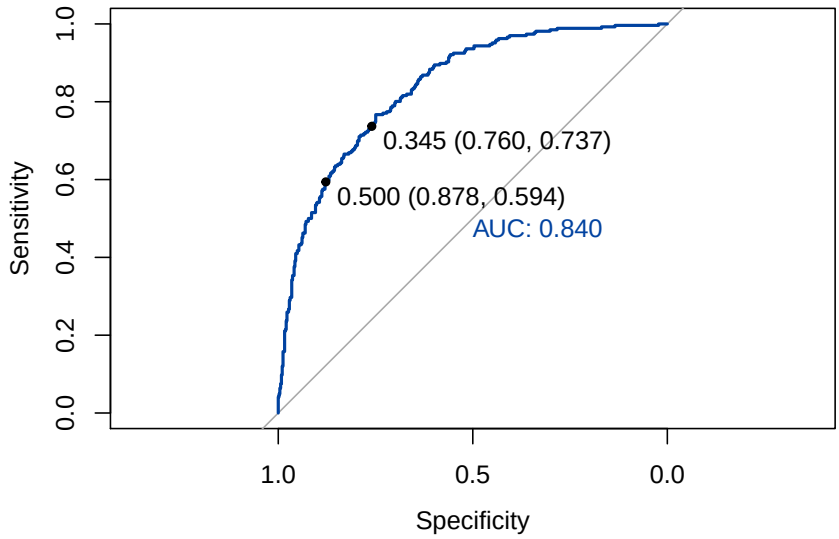
A receiver operating characteristic (ROC) curve is a graphical representation of the trade-offs between type-1 (false positive) and type-2 (false negative) errors. AUC (area under curve) is the summary of how the model performs at different decision thresholds.

In diagnostic medicine, testing the hypothesis that the ROC Curve area or partial area has a specific value is a common practice[12].

Our model has AUC of 84%. This is a reasonably good value.

[12] *Statistical Methods in Diagnostic Medicine*, Second Edition, Ch.4, Author(s): Xiao-Hua Zhou Nancy A. Obuchowski Donna K. McClish

```
plot(pROC::roc(diabetes %>% filter(BMI != 0) %>%
               pull(Diagnosis), fitted(fit.f)),
     col = "#0042A0",
     print.thres = c(0.5, 0.345), print.auc = T)
```



Sensitivity and Specificity

Sensitivity measures the proportion of correct forecast of presence of diabetes (true positive rate) w.r.t. to all positive diagnosis, while specificity measures the proportion of correct forecast of absence of diabetes (true negative rate) w.r.t. all negative diagnosis.

At the cut-off point of 0.5 for fitted values (refer to ROC curve above), the sensitivity is 88% and specificity is 59%. By adjusting the cut-off point for positive diagnosis to 0.345, a sensitivity of 76% and specificity of 74% can be achieved.

Comparison with Smith et al.

Smith et al. describe their methodology as follows:

> "Once the algorithm had been trained using 576 cases, ADAP was used to forecast whether another 192 test cases would develop diabetes

within five years. Forcing ADAP to conclude on all test cases produced a sensitivity and specificity of 76 percent. A receiver operating characteristic (ROC) curve was determined."

In order to make a direct comparison with Smith et al., data is split into two groups: a training set with sample size of 576 and a test set with the remaining 192 observations. Model is trained using training set (new coefficients of logistic regression are calculated) and forecasting performance of the trained model on the test set is assessed in terms of error rate, sensitivity and specificity. This procedure is repeated 100 times and summary statistics are presented for error rate, sensitivity and specificity.

Smith et al. report sensitivity, specificity, and ROC curve when ADAP was applied on all test cases. Logistic model's sensitivity, specificity, and ROC curve, when regressed on all test cases, has already been presented in the earlier sections. Smith et al. did not report the error rate of their learning algorithm.

The forecasting evaluation procedure is presented below:

Define helper functions

Define a function to split data randomly into training set of 576 observations and data set of remaining 192 observations.

```
# Split data (randomly) into training and
#   test data sets.
split_data <- function(sample_size = 576) {
  out <- list()
  out$training <- diabetes %>%
    mutate(id = row_number()) %>%
    sample_n(sample_size) %>%
    filter(BMI != 0)
  out$test <- diabetes %>%
    mutate(id = row_number()) %>%
    anti_join(out$training, by = 'id') %>%
    filter(BMI != 0)
  return(out)
}
```

Define a prediction function. Using training set, calculate the coefficients of the regressors. Use the trained model on the test data to predict the outcome of diabetes diagnosis.

```
# Fit the model on the training data, validate on
#   test data, and return predicted results.
predict_diabetes <- function(training_data, test_data) {
  formula <- Diagnosis ~ Glucose + log(BMI) +
    Pregnancies + DPF + Age + DPF:Glucose
```

```
  fit.model <- glm(formula, training_data,
                   family = binomial(link = "logit"))
  shadow_val <- predict(fit.model,
                        newdata = test_data)
  return(arm::invlogit(shadow_val))
}
```

Error rate

Compare the predicted outcome to data (truth) and report error rate. The procedure is repeated 100 times to reveal the distribution characteristics.

```
validate_error_rate <- function(threshold) {
  return(c(1:100) %>%
           map_dbl(~ {
             splitted <- split_data()
             predicted <- predict_diabetes(splitted$training,
                                           splitted$test)
             error_rate(predicted, splitted$test$Diagnosis,
                        threshold = threshold)
           }))
}
result <- validate_error_rate(threshold = 0.5)
summary(result)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17.37   21.58   23.14   23.09   25.03   28.42
```

At predictor threshold of 0.5 for positive values, the average error rate is 23.1%.

Predictor performance depends on the threshold for deciding positive diagnosis. A threshold of 0.35 (instead of 0.5) improves specificity at the expense of sensitivity. Average error rate is also slightly higher.

```
summary(validate_error_rate(threshold = 0.35))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.95   23.61   25.46   25.10   26.74   32.98
```

Sensitivity

Sensitivity is the ability of the model to correctly forecast diabetes (true positive rate). As before, learning procedure is repeated 100 times and sensitivity is calculated for each iteration.

```
sensitivity <- c(1:100) %>%
  map_dbl(~ {
    splitted <- split_data()
```

```
    predicted <- predict_diabetes(splitted$training,
                                  splitted$test)
    caret::sensitivity(
      factor(if_else(predicted >= 0.35, 1, 0)),
      factor(splitted$test$Diagnosis))
  })
summary(sensitivity)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.6825  0.7355  0.7520  0.7566  0.7841  0.8843
```

At 0.35 threshold for deciding positive diagnosis we obtain average sensitivity of 75.7%.

Specificity

Specificity is the ability of the model to correctly forecast absence of diabetes after 5 years (true negative rate). Learning procedure is repeated 100 times and specificity is calculated for each iteration.

```
specificity <- c(1:100) %>%
  map_dbl(~ {
    splitted <- split_data()
    predicted <- predict_diabetes(splitted$training,
                                  splitted$test)
    caret::specificity(
      factor(if_else(predicted >= 0.35, 1, 0)),
      factor(splitted$test$Diagnosis))
  })
summary(specificity)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5833  0.6883  0.7381  0.7295  0.7684  0.8596
```

At 0.35 threshold for deciding positive diagnosis we obtain average sensitivity of 72.9%.

Conclusions

The performance of the model depends on the cut off point used to define a positive diagnosis. At 0.5 there are less false positives but more false negatives. At 0.35 false positives roughly equal false negatives. Less false positives may be preferable to less false negatives, since false positive results can harm the subject emotionally or financially.

Blood glucose level has a significant influence in the diagnosis of diabetes. However, incorporating other relevant risk factors and their interaction improves the predictive power of the model.

Although logistic regression compares favorably with other techniques, such as the one used by Smith et al., limitations of data and

inaccuracies present in observations prevent the model from realizing its full potential.

Appendix

The R markdown file, *Report.Rmd*, used for generating this report can be found at github.

Url for the data set is embedded in the markdown file. No additional resources are necessary to generate the report.