Sta141a Final Report

12/9/2019

Collaborators:

Grant Smith: EDA (normality), SVM,KNN,Discussion (Q1),Github

Isaiah Valencia: EDA(linearity),Log Regr, Interpretation (Q3-6)

Hun Lee: EDA(correlation),Log Regr,RF,Discussion (Q2,3-6)

Christopher Ton: EDA(PCA),RF,Discussion(Q1,Q5), Github

Overview of the problem

We aim to predict whether or not a Pima Indian Woman is diabetic or not based on her health characteristics such as pregnancy, glucose, blood pressure, skin, bmi, pedigree, and age. Additionally, we will elaborate on which biological features (i.e. glucose levels) are best suited in correctly identifying diabetes in the subject group. Throughout this analysis we will look at accuracy, sensitivity, and specificity. While we want all of these values to be high, some analysis will emphasize sensitivity over specificity. Due to the medical nature of the data we believe importance should be placed on correctly identifying an individual with the disease as correct and accurate diagnosis plays an important role in proper treatment of patients.

Exploratory Data Analysis

We first calculated summary statistics for the dataset in order to obtain a clearer picture of the variables we are working with.

(Data)

Summary Statistics

The data set we are using comes from the package 'mass' and consists of the variables npreg, glu, bp, skin, bmi, ped, and age. Type/Class is a binary variable indicating whether a subject has or does not have diabetes. In our original proposal the variable 'insulin' was to be included in the dataset, however due to the high number of missing values we chose to drop this variable from the project.

```
##      npreg             glu              bp              skin
##  Min.   : 0.000   Min.   : 56.00   Min.   : 24.00   Min.   : 7.00
##  1st Qu.: 1.000   1st Qu.: 98.75   1st Qu.: 64.00   1st Qu.:22.00
##  Median : 2.000   Median :115.00   Median : 72.00   Median :29.00
##  Mean   : 3.517   Mean   :121.03   Mean   : 71.51   Mean   :29.18
##  3rd Qu.: 5.000   3rd Qu.:141.25   3rd Qu.: 80.00   3rd Qu.:36.00
##  Max.   :17.000   Max.   :199.00   Max.   :110.00   Max.   :99.00
##       bmi             ped              age          type
##  Min.   :18.20   Min.   :0.0850   Min.   :21.00   No :355
##  1st Qu.:27.88   1st Qu.:0.2587   1st Qu.:23.00   Yes:177
##  Median :32.80   Median :0.4160   Median :28.00
##  Mean   :32.89   Mean   :0.5030   Mean   :31.61
##  3rd Qu.:36.90   3rd Qu.:0.6585   3rd Qu.:38.00
##  Max.   :67.10   Max.   :2.4200   Max.   :81.00
```

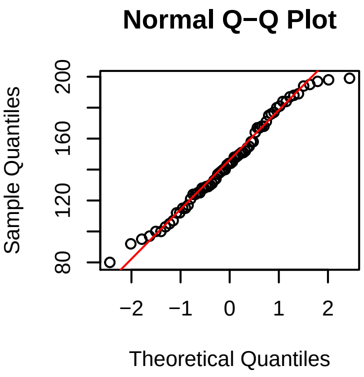
Additionally, we split our data set into training and test sets with a ratio of 70:30. The training set consists of 372 observations and the test set consists of 160 observations.

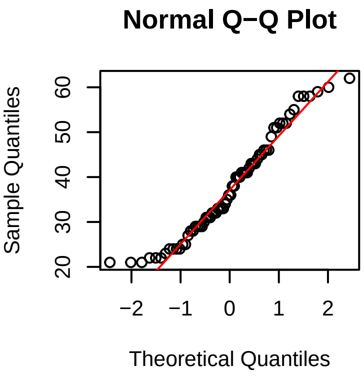
Data Analysis

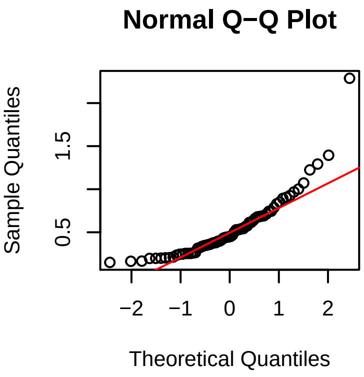
Assumptions

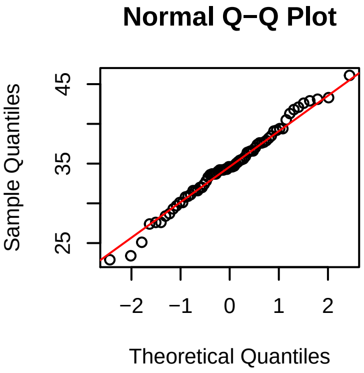
Normality Results:

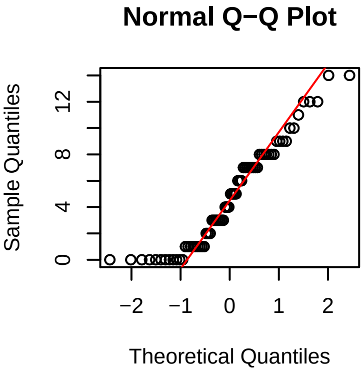
Initial qq plots for the key variables indicated that normality violations would be an issue regarding this dataset. The plots below for example represent glu, age, ped, and bmi











```
##    glu    age    ped    bmi  npreg     bp   skin
## 0.2532 0.0099 0.0000 0.8835 0.0004 0.3545 0.0000
```

The shapiro wilks test confirms that only bp, bmi, and glu are normally distributed. (using 0.01 alpha)

```
##   glu   age   ped   bmi npreg    bp  skin
## 0.007 0.000 0.000 0.100 0.000 0.441 0.006
```

For the group negative for diabetes, we can see from the shapiro wilks test that skin, glu, age, ped, and npreg all violate the normality assumption. Only bp and bmi are normal. (using .01 alpha)

Conclusion from normality tests

The only two variables that exhibit normality based on the shapiro wilks test for both yes and no diabetes groups are bp and bmi. Therefore we intend to focus on methods that do not rely on the assumption of normality.

Methodology

(1) KNN

KNN only requires that our data be from a random sample. Additionally we will standardize our dataset before carrying out our analysis. The data has been split with 70% of the data being alloted for training and the remaining 30% being used for testing. It is difficult to gain any inference from KNN so we will mostly be using it to confirm other methods conclusions regarding which variables are best for classifying diabetes.

(2) SVM

SVM will be used to answer which variables will be most useful in predicting diabetes. We initially used SVM with all the predictors. We then used SVM feature selection to determine which variables will be most useful in correcly classifying diabetes. The results were then compared used both linear and radial kernals.

(3) PCA

```
## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5     PC6     PC7
## Standard deviation     1.5163 1.2438 1.0041 0.9027 0.8405 0.56395 0.55345
## Proportion of Variance 0.3285 0.2210 0.1440 0.1164 0.1009 0.04543 0.04376
## Cumulative Proportion  0.3285 0.5495 0.6935 0.8099 0.9108 0.95624 1.00000
```

We considered reducing the dimensions of our data by finding candidate variables contributing in direction of the most variance, thus ignoring any of the unnecessary noise in our data. Particularly, we wished to maximize the amount of information along these directions, or namely, the eigenvectors of our dimensions. PC1,PC2,PC3,PC4,PC5 collectively describe at least 90% of the variance. However, due to the lack of high correlation among the variables, PCA fails to determine the minimal number of components to account for the variability and thus, most, if not all, of the predictors may be necessary for the determination of diabetes.

An effective approach with PCA would then pursue LDA, or perhaps QDA based on the determined principal components. To meet the assumptions of discriminant analysis, we tested to see if normality holds. As the above results show, our data does not meet the assumption of normality.

(4) RF

RF considers all predictors,fits 1000 bootstrapped decision trees and returns the list of the variables with statistical information of the importance, including mean decrease accuracy and mean decrease gini. Respectively, these values measure accuracy and node purity reduction. Generally, larger values imply greater importance. Accuracy measures based on the confusion matrix were made for the OOB dataset RF does not use to train and our intially defined 70/30 testing set.

(5) Logistic Regression

Multiple logistic regression was used to mainly answer some of the questions about the variable's effect on diabetes and potential interactions. Summary results such as the significance and AIC were assessed to determine how diabetes is predicted.

(6) Unused Methods

Within our proposal we originally planned to use LDA/QDA and Bayes Naive Classifiers in our analysis. Upon testing for normality we eliminated LDA/QDA from potential use due to the normality assumption violation. For Bayes Naive Classifiers, our research shows that BNC doesn't work well with interacting features and there was little to gain in terms of statistical inference.

Question 1: *What is the best combination of variables in predicting diabetes in Pima Indians?*

An intial method we utilized was SVM for determining which variables are best suited to determine Diabetes. The full svm model was calculted and then tuned to increase performance.

The initial model gave the following results:

```
## [[1]]
##
## svm.fullpr No Yes
##        No  98  25
##        Yes 18  29
##
## [[2]]
##       Accuracy Sensitivity Specificity
## [1,] 0.7470588    0.537037   0.8448276
```

Tuning the model in order to further optomize the results yieled a model with improved error and dispersion. Under this model the below results captured the accuracy, sensitivity and specificity when predicting diabetes using SVM with all features. Upon reviewing these results we then proceeded to use feature selection through SVM to improve accuracy. Svm would consistently include glu,bmi,ped, and npreg as the best features to use for maximizing results. On occasion an additional predictor would be included in the model but the lack of consistency lead to our decision to only focus on the features that occured in every iteration of the experiment.

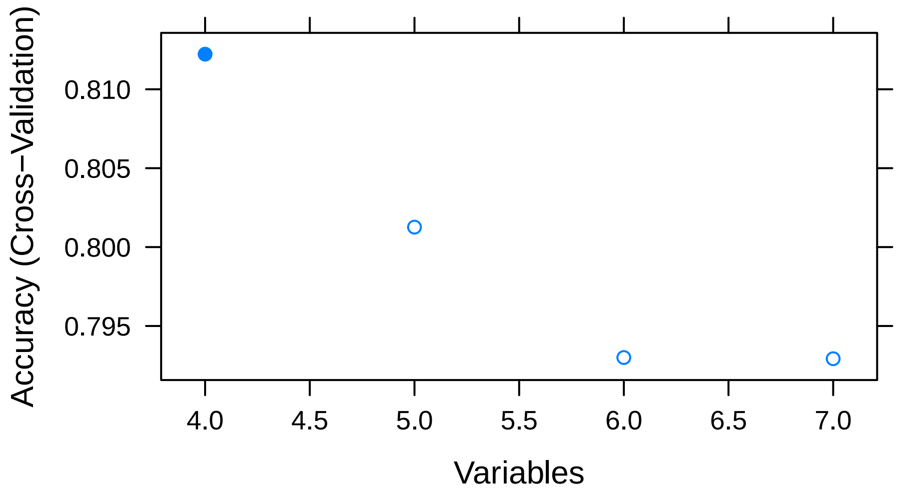
Even with the best features used in the svm model, we saw little performance increase. The next step we took was to attempt to use different kernal's to see if results improved. We initially used a linear kernal for both the full and feature selection model. Results saw little to no improvement. Howver, once we applied the kernal change to radial for the reduced model using only the selected features, we saw accuracy and specificty both improve.

```
## 
## Parameter tuning of 'svm':
## 
## - sampling method: 10-fold cross validation 
## 
## - best parameters:
##  cost
##  0.01
## 
## - best performance: 0.2044294 
## 
## - Detailed performance results:
##    cost     error dispersion
## 1 1e-03 0.3399399 0.09129766
## 2 1e-02 0.2044294 0.07469544
## 3 1e-01 0.2126877 0.06208229
## 4 1e+00 0.2071321 0.05835227
## 5 5e+00 0.2071321 0.05835227
## 6 1e+01 0.2071321 0.05835227
```

```
## [[1]]
##      
## tn.ts  No Yes
##   No  103  28
##   Yes  13  26
## 
## [[2]]
##       Accuracy Sensitivity Specificity
## [1,] 0.7588235   0.4814815    0.887931
```

select features

```
## 
## Recursive feature selection
## 
## Outer resampling method: Cross-Validated (10 fold) 
## 
## Resampling performance over subset size:
## 
##  Variables Accuracy  Kappa AccuracySD KappaSD Selected
##          4   0.8122 0.5534    0.07582  0.1928        *
##          5   0.8013 0.5293    0.07470  0.1917         
##          6   0.7930 0.5074    0.05983  0.1608         
##          7   0.7929 0.5057    0.05711  0.1540         
## 
## The top 4 variables (out of 4):
##    glu, bmi, ped, npreg
```



SVM with selected features

The following results use glu, bmi, ped, and npreg as predictors.

```
##
## svm.pr2 No Yes
##     No  96  26
##     Yes 20  28
```

```
##       Accuracy Sensitivity Specificity
## [1,] 0.7294118   0.5185185   0.8275862
```

Change Kernal

Using the full model with a radial kernel, we only saw specificity increase slightly but accuracy and sensitivity did not improve.

```
##       Accuracy Sensitivity Specificity
## [1,] 0.7176471         0.5   0.8189655
```

Using the reduced model with a radial kernel yieled the following results.

```
##       Accuracy Sensitivity Specificity
## [1,] 0.7352941         0.5   0.8448276
```

Summary Results

```
## $`Full Linear`
##
## svm.fullpr No Yes
##        No  98  25
##        Yes 18  29
##
## $`Reduced Linear`
##
## svm.pr2 No Yes
##     No  96  26
##     Yes 20  28
##
## $`Full Radial`
##
## svm.pr4 No Yes
##     No  95  27
##     Yes 21  27
##
## $`Reduced Radial`
```

```
##
## svm.pr3 No Yes
##     No  98  27
##     Yes 18  27
```

```
## $`Full Linear`
##       Accuracy Sensitivity Specificity
## [1,] 0.7470588    0.537037   0.8448276
##
## $`Reduced Linear`
##       Accuracy Sensitivity Specificity
## [1,] 0.7294118   0.5185185   0.8275862
##
## $`Full Radial`
##       Accuracy Sensitivity Specificity
## [1,] 0.7176471         0.5   0.8189655
##
## $`Reduced Radial`
##       Accuracy Sensitivity Specificity
## [1,] 0.7352941         0.5   0.8448276
```

KNN

Full KNN

Using Knn with all variables yields the following results which perform similarly to SVM.

```
##
## pm.knn12  No Yes
##      No  101  22
##      Yes  15  32
```

```
## [[1]]
##           [,1]      [,2]      [,3]
## [1,] 0.7823529 0.5925926 0.8706897
```

KNN:Selected Features

Using knn with the selected variables from SVM performed higher than using KNN with all features. This further supports SVM's conclusion that using selected features will yield at least as good of a prediction. In regards to selecting K for this method, we used an accuracy optimizer and then cross validation to determine which value of K would yield the best results. These tools are featured in the appendix.

```
##
## pm.knn12 No Yes
##      No  97  25
##      Yes 19  29
```

```
## [[1]]
##           [,1]     [,2]      [,3]
## [1,] 0.7411765 0.537037 0.8362069
```

note: While the optimizer would yield diffent results due to randomness in tie breaking, it still gave us a ballpark k that could then be used with cross-validation. This allowed us to save computing power by not testing an excessive number of K-values.

```
## [1] 40
```

```
##   k
## 5 5
```

Random Forest

Classification based on all scaled predictors, assessment for OOB set

```
##      No Yes class.error
## No  214  25   0.1046025
## Yes  46  77   0.3739837
```

```
##      Accuracy Sensitivity Specificity
## [1,] 0.802806    0.754902   0.8230769
```

Assessment for splitted scaled testing set

```
##      pred
##       No Yes
##   No  92  24
##   Yes 24  30
```

```
##       Accuracy Sensitivity Specificity
## [1,] 0.7176471   0.5555556   0.7931034
```

```
##              No       Yes MeanDecreaseAccuracy MeanDecreaseGini
## npreg 13.355495  2.764417            12.617932         13.50547
## glu   46.969702 50.578823            61.043950         50.35071
## bp    -0.964779 -4.767178            -3.571863         12.70995
## skin   3.665803  9.136831             8.817524         15.51248
## bmi    4.566463 17.562843            15.873631         24.23137
## ped   10.007798 10.665396            13.891581         23.44552
## age   19.177595 14.556106            24.026034         22.23082
```

The accuracy score for both testing sets are identical, indicating that RF is a consistent classifier.

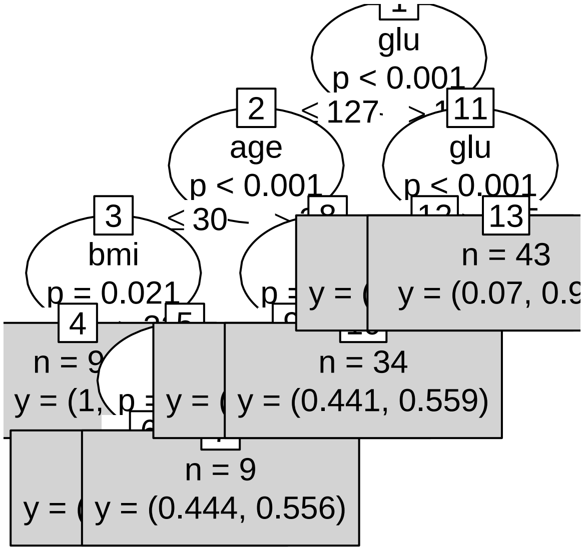
We try to remove any outliers (although RF is insensitive to it), to see if improvements can be made.

```
##
## rf.1.pred No Yes
##       No  88  20
##       Yes 14  33
```

```
## [1] 0.7806452
```

Small improvement in prediction may not be significant enough to support the exclusion of outliers.

** Illustration of Decision Tree **



For illustration of the process, we consider a single decision tree,produced from the package Rpart, that analyzes 5 variables: glu, ped, age, bmi and npreg.The tree finds a cutoff point for a particular variable that increases entropy in the system. For instance, glucose levels would be the greatest dividing factor. The process continues to divide the separated groups based on the next variable. A random forest algorithm does this at random and repeatedly for a specified number of trees and the final decision is determined as whole by count of the terminal nodes.

Question 2: *Is there any interaction effect between the interested variables?*

```
## Pairing glu with age
## Pairing glu with bmi
## Pairing glu with npreg
## Pairing glu with ped
## Pairing glu with skin
## Pairing glu with bp
## Pairing age with bmi
## Pairing age with npreg
## Pairing age with ped
## Pairing age with skin
## Pairing age with bp
## Pairing bmi with npreg
## Pairing bmi with ped
## Pairing bmi with skin
## Pairing bmi with bp
## Pairing npreg with ped
## Pairing npreg with skin
## Pairing npreg with bp
## Pairing ped with skin
## Pairing ped with bp
## Pairing skin with bp
##
##                                Method: vimp
##                      No. of variables: 7
##             Variables sorted by VIMP?: TRUE
##     No. of variables used for pairing: 7
##      Total no. of paired interactions: 21
##              Monte Carlo replications: 3
##      Type of noising up used for VIMP: permute
##
##             Var 1   Var 2 Paired Additive Difference
## glu:age    0.0903  0.0289 0.1350   0.1192     0.0157
## glu:bmi    0.0903  0.0121 0.1105   0.1024     0.0081
## glu:npreg  0.0903  0.0084 0.1044   0.0987     0.0056
## glu:ped    0.0903  0.0060 0.1099   0.0963     0.0135
## glu:skin   0.0903  0.0016 0.0979   0.0919     0.0060
## glu:bp     0.0903 -0.0022 0.0902   0.0882     0.0021
## age:bmi    0.0274  0.0125 0.0473   0.0399     0.0074
## age:npreg  0.0274  0.0086 0.0379   0.0360     0.0019
## age:ped    0.0274  0.0069 0.0411   0.0342     0.0069
## age:skin   0.0274  0.0010 0.0304   0.0284     0.0019
## age:bp     0.0274 -0.0013 0.0291   0.0261     0.0031
## bmi:npreg  0.0123  0.0084 0.0250   0.0207     0.0043
## bmi:ped    0.0123  0.0064 0.0197   0.0187     0.0010
## bmi:skin   0.0123  0.0019 0.0147   0.0142     0.0005
## bmi:bp     0.0123 -0.0022 0.0104   0.0101     0.0003
## npreg:ped  0.0070  0.0071 0.0171   0.0140     0.0030
## npreg:skin 0.0070  0.0017 0.0109   0.0087     0.0022
## npreg:bp   0.0070 -0.0013 0.0056   0.0057    -0.0001
## ped:skin   0.0087  0.0025 0.0101   0.0112    -0.0010
## ped:bp     0.0087 -0.0026 0.0049   0.0061    -0.0011
## skin:bp    0.0033 -0.0018 0.0008   0.0014    -0.0006
```

```
##
##                                Method: maxsubtree
##                      No. of variables: 7
##   Variables sorted by minimal depth?: TRUE
##
##         glu  age  bmi  ped npreg skin   bp
## glu    0.07 0.18 0.18 0.18  0.22 0.21 0.25
## age    0.17 0.11 0.20 0.20  0.28 0.23 0.27
## bmi    0.21 0.24 0.15 0.24  0.30 0.28 0.30
## ped    0.22 0.27 0.23 0.18  0.28 0.27 0.27
## npreg  0.27 0.32 0.29 0.30  0.19 0.35 0.35
## skin   0.29 0.33 0.31 0.30  0.39 0.21 0.36
## bp     0.37 0.41 0.38 0.39  0.49 0.42 0.27
```

Considering there is a high correlation between “npreg” and “age”, it does make sense that these two variables has the highest interaction. The interesting result is that the interaction between “ped” and “npreg” has the second highest interaction as expected because we were able to confirm that the interaction effect between “ped” and “npreg” contributes to a positive result in increasing accuracy in both K-fold and logistic regression methods.

Question 3: *Is there a relationship between blood pressure and diabetes?*

Comparison of logistic model with & without bp:

```
## glm(formula = type ~ glu + bmi + ped + age, family = binomial(link = logit),
##     data = pima.train)
```

```
##               True type
## Predicted type No Yes
##            No  94  12
##            Yes 15  34
```

```
## [[1]]
##           [,1]      [,2]      [,3]
## [1,] 0.8258065 0.7391304 0.8623853
```

```
## glm(formula = type ~ glu + bmi + ped + age + bp, family = binomial(link = logit),
##     data = pima.train)
```

```
##               True type
## Predicted type No Yes
##            No  94  11
##            Yes 15  35
```

```
## [[1]]
##           [,1]      [,2]      [,3]
## [1,] 0.8322581 0.7608696 0.8623853
```

We created two logistic regression models: one with “glu” + “bmi” + “ped” + “age” + “bp” and the other without blood pressure, “bp”. As the model summary indicates, we are able to see that all the independent variables are significant except “bp” in the model with “glu” + “bmi” + “ped” + “age” + “bp” whereas dropping “bp” variable not only decreases AIC value and gives all significant independent variables in the logistic model.

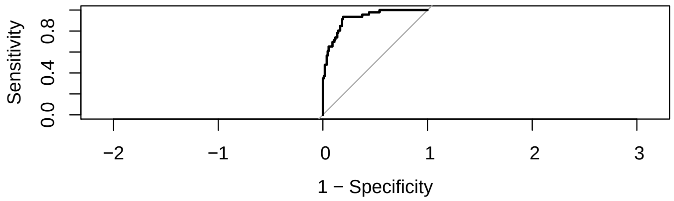
*Purposeful Selection: comparing likelihood ratio tests for models

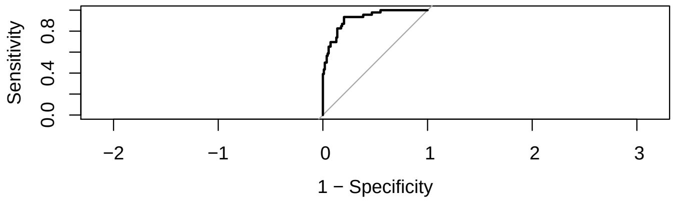
```
## Analysis of Deviance Table
##
## Model 1: type ~ glu + bmi + ped + age
## Model 2: type ~ glu + bmi + ped + age + bp
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       357     292.16
## 2       356     291.56  1  0.60143    0.438
```

When we conducted the likelihood-ratio test for the goodness of fit of two models based on the ratio of their likelihoods under the null hypothesis: the first model = the first model with “bp”, we were not able to significant evidence to reject the Ho and hence we concluded the variable “bp” is not needed for the diabetes prediction model.

K-Fold Cross Validation

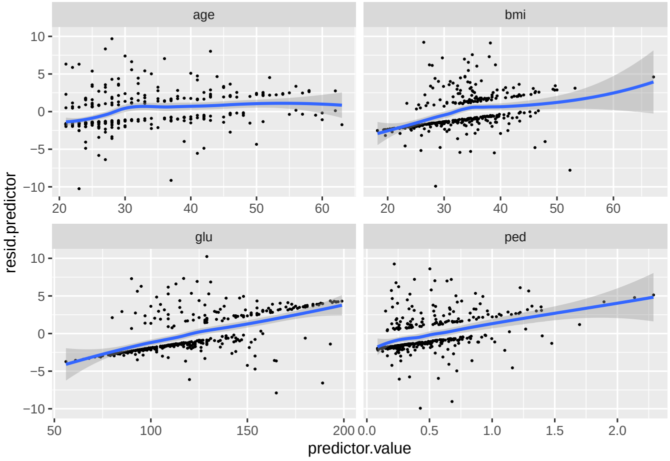
```
## Setting levels: control = No, case = Yes

## Setting direction: controls < cases

## Setting levels: control = No, case = Yes

## Setting direction: controls < cases

## Area under the curve: 0.9192

## Setting levels: control = No, case = Yes
## Setting direction: controls < cases

## Setting levels: control = No, case = Yes

## Setting direction: controls < cases

## Area under the curve: 0.9192
```

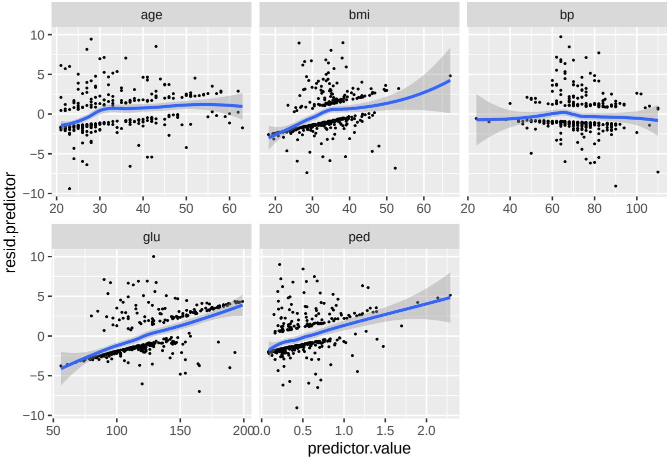




Please refer to below for interpretation on ROC plots.

```
##      Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
## -2.96021 -0.46397 -0.21435 -0.02298  0.40345  2.98123

##      Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
## -2.80642 -0.45698 -0.20972 -0.02236  0.39634  2.94144
```





Question 4: *How does pedigree effect pregnant women and diabetes?*

```
## glm(formula = type ~ glu + bmi + ped + age, family = binomial(link = logit),
##     data = pima.train)
```

```
## [1]  0.0000  1.0431  1.1075 10.0033  1.0588
```

```
## Waiting for profiling to be done...
```

```
##               2.5 %  97.5 %
## (Intercept) 0.0000  0.0000
## glu         1.0319  1.0554
## bmi         1.0586  1.1631
## ped         3.8425 27.5715
## age         1.0296  1.0900
```

Please refer to below for intepretation on effects.

Question 5: *Is there a relationship between diabetes and pregnancy?*

```
## [[1]]
## glm(formula = type ~ npreg, family = binomial(link = logit),
##     data = pima.train)
##
## [[2]]
## [1] 446.3501
```

```
## [[1]]
## glm(formula = type ~ age, family = binomial(link = logit), data = pima.train)
##
## [[2]]
## [1] 431.2967
```

```
## [[1]]
## glm(formula = type ~ npreg + age, family = binomial(link = logit),
##     data = pima.train)
##
## [[2]]
## [1] 432.6867
```

Both predictors, age and npreg, two of the most correlated variables, may not be necessary for determining diabetes. For an instance we ran, in descending order of AIC values, 441.71,423.65, and 422.87, we see the (type~ npreg) performs worst compared (type~age) which is best. In the model that considers both predictors, npreg is deemed insignicant. In terms of choosing which variable is better for prediction accuracy between "age" and "npreg", we prefer age to pregnancy because not only does the model with "age" give lower AIC, but also it tends to give higher prediction accuracy.

Overall Intepretation and Discussions for Questions 4

Effect of Variables on odds of Diabetes in Pima Women

The interpretation of the variables included in the model will be understood as the effect of the variable on the odds (not log odds) of diabetes for female pima indian women.That is, for each unit increase in glucose, bmi, pedigree, and age the effects is an increase in odds of diabetes for Pima indian women. This effect is the exp() for each parameter coefficient, which can be seen. Since this is a random variable, a 95% confidence interval is provided below for effects. Given that interpretation is important, each variable will be explained at length. One unit increase in Glucose increases the odds of diabetes at by at least 3.1 percent and at most 5.5 percent. One unit increase in BMI increases the odds by at least 6.3 percent to at most 17 percent, hold all other variables constant. One unit increase in the pedigree function increases the odds by at least 2.44 times to at most 15 times as much, thus the most contributing variable in the model. Lastly, a one unit increase in age increased odd by at least 4.5 percent to at most 11 percent. This can also be interpreted in probability and not odds, as previously stated. That is, since the sign of all exponentiated variables are positive, as a single variable increases, holding all others constant, so to does the probability of diabetes in Pima indian women.

ROC and predictic

Next, consider the receiver operating characteristic (ROC). We turn the attention to the plot because of the drawbacks of the classification table. Classification, in our sense, classifies or predicts an observation based on the predicted value and a threshold. This being a predicted value that is a probability and a probability thresholdthreshold. If the predicted probability is above, say 0.5 (threshold), take that as evidence that subject has diabetes, say. Taking the findings for all observations can be summarized in a classification table. But a classification table is reduces results from its continuous probability form, to a binary form, so information is lost to summary, therefore a drawback. Moreover, classification is assessed by accuracy = P(correct classification) = sensitivity * P(y = 1) + specificity * (1 - P(y = 1)). Therefore accuracy depends on the relative number of times that y = 1 or y = 0, in our case, the number of subjects that have diabetes and those that don’t, this being the second drawback. The ROC addresses these drawbacks. For the ROC curve, the threshold = 0 results in diabetes classification, therefore sensitivity = 1 and for threshold = 1 results in non diabetes, therefore sensitivity = 0 and 1 - specificity specificity = 1. Referring to the graph, for any input of threshold or any input of specificity, the corresponding sensitivity can be found. For some specificity value, Higherhigher sensitivity means greater predictive power, thus greater area is greater predictive power. Our model suggest reaches an area of 0.906, near the max being 1. The area, known as concordance index, is the probability that predictions are Concordant with observations.

R Appendix

```r
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
library(ggplot2)
library(plotly)
library(dplyr)
#library(ggbiplot)
library(MASS)
library(ISLR)
library(boot)
library(class)
library(caret)
library(randomForest)
library(randomForestSRC)
library(car)
#install.packages('e1071', dependencies=TRUE)
#install.packages('kernlab')
library(e1071)
library(kernlab)
library(caret)
library(pROC)
library(party)
data("Pima.te")

#training and test data (70:30) split
Pim.N<-as.data.frame(rbind(Pima.te,Pima.tr))
trng = sample(1:532, 362, replace=FALSE)
test.g<- setdiff(1:532,trng)

Pim.tr<-Pim.N[trng,]
Pim.te<-Pim.N[test.g,]

#Standardize the varaibles being used
pm.tr.sc<-data.frame(scale(Pim.tr[-8],center = TRUE,scale=TRUE))
pm.tr.sc<-data.frame(cbind(pm.tr.sc),Class=Pim.tr[,8])

pm.ts.sc<-data.frame(scale(Pim.te[-8],center = TRUE,scale=TRUE))
pm.ts.sc<-data.frame(cbind(pm.ts.sc),Class=Pim.te[,8])
summary(Pim.N)

pos.db<-subset(Pima.tr, type=='Yes')
par(mfrow=c(2,3))
for (i in c('glu','age','ped','bmi','npreg'))
{
  qqnorm(pos.db[[i]]);qqline(pos.db[[i]],col=2)
}
pp.st<-numeric(0)
for (i in c('glu','age','ped','bmi','npreg','bp','skin'))
{
s.test<-shapiro.test(pos.db[[i]])
pp.st[i]<-s.test$p.value
}
round(pp.st,4)
#qqnorm plots for 'No' class
neg.db<-subset(Pima.tr, type=='No')

# par(mfrow=c(3,3))
# for (i in c('glu','age','ped','bmi','npreg','bp','skin'))
# {
# qqnorm(neg.db[[i]]);qqline(neg.db[[i]],col=2)
# }
#shapiro wilkes test for normality
np.st<-numeric(0)
for (i in c('glu','age','ped','bmi','npreg','bp','skin'))
{
s.test<-shapiro.test(neg.db[[i]])
np.st[i]<-s.test$p.value
}
round(np.st,3)
pima.pca <- prcomp(Pim.tr[,-8], center = TRUE,scale = TRUE)
summary(pima.pca) # PCI explains 34%,PC2 explains 20%

#cor(pima[,-8])


#svm full model
#>>>>>>> a069b6d0479eefafb5ea061973707b23cfb9c562
```

```
svm.md<-svm(Class~.,data=pm.tr.sc,type= 'C-classification', kernel='linear')
#summary(svm.md)

svm.fullpr<-predict(svm.md,pm.ts.sc)
svm.fm<-table(svm.fullpr,pm.ts.sc$Class)

svmfm.res<-cbind(sum(diag(svm.fm))/sum(svm.fm),#accuracy
svm.fm[4]/sum(svm.fm[,2]), #sensitivity
svm.fm[1]/sum(svm.fm[,1])) #specificity
colnames(svmfm.res)<-c('Accuracy','Sensitivity','Specificity')
list(svm.fm,svmfm.res)
#svm full model
lin.tn<-tune.svm(Class~.,data=pm.tr.sc,kernel='linear',cost=c(0.001,0.01,0.1,1,5,10))
summary(lin.tn)
#pick model
svm.ln.best<-lin.tn$best.model

tn.ts<-predict(svm.ln.best,newdata=pm.ts.sc)
svm.tn.cm<-table(tn.ts,pm.ts.sc$Class)

ln.res<-cbind(sum(diag(svm.tn.cm))/sum(svm.tn.cm),#accuracy
svm.tn.cm[4]/sum(svm.tn.cm[,2]), #sensitivity
svm.tn.cm[1]/sum(svm.tn.cm[,1])) #specificity
colnames(ln.res)<-c('Accuracy','Sensitivity','Specificity')
list(svm.tn.cm,ln.res)
set.seed(2)
feat<-rfeControl(functions=lrFuncs, method="cv", number=10)
svm.feat <- rfe(pm.tr.sc[,1:7], pm.tr.sc[,8],sizes = c(7, 6, 5, 4),
              rfeControl = feat, method = "svmLinear")

svm.feat
plot(svm.feat)

svm.md2<-svm(Class~glu+bmi+ped+npreg,pm.tr.sc,type="C-classification",kernel='linear')
svm.pr2<-predict(svm.md2,pm.ts.sc[,c(1,2,5,6)])

(svm.cm2<-table(svm.pr2,pm.ts.sc[,8]))
res.sv2<-cbind(sum(diag(svm.cm2))/sum(svm.cm2),
svm.cm2[4]/sum(svm.cm2[,2]), #sensitivity
svm.cm2[1]/sum(svm.cm2[,1])) #specificity

colnames(res.sv2)<-c('Accuracy','Sensitivity','Specificity');res.sv2
svm.md4<-svm(Class~.,pm.tr.sc,type="C-classification",kernel='radial')
svm.pr4<-predict(svm.md4,pm.ts.sc[,-8])

svm.cm4<-table(svm.pr4,pm.ts.sc[,8])
res.sv4<-cbind(sum(diag(svm.cm4))/sum(svm.cm4),
svm.cm4[4]/sum(svm.cm4[,2]), #sensitivity
svm.cm4[1]/sum(svm.cm4[,1])) #specificity

colnames(res.sv4)<-c('Accuracy','Sensitivity','Specificity');res.sv4
svm.md3<-svm(Class~glu+bmi+ped+npreg,pm.tr.sc,type="C-classification",kernel='radial')
svm.pr3<-predict(svm.md3,pm.ts.sc[,c(1,2,5,6)])

svm.cm3<-table(svm.pr3,pm.ts.sc[,8])
```

```r
res.sv3<-cbind(sum(diag(svm.cm3))/sum(svm.cm3),
svm.cm3[4]/sum(svm.cm3[,2]), #sensitivity
svm.cm3[1]/sum(svm.cm3[,1])) #specificity

colnames(res.sv3)<-c('Accuracy','Sensitivity','Specificity');res.sv3
svm.mats<-list(svm.fm,svm.cm2,svm.cm4,svm.cm3)
svm.results<-list(svmfm.res,res.sv2,res.sv4,res.sv3)
theSVM<-c('Full Linear','Reduced Linear','Full Radial','Reduced Radial')
theSVM2<-c('Full Linear','Reduced Linear','Full Radial','Reduced Radial')
names(svm.results)<-theSVM;names(svm.mats)<-theSVM2

svm.mats;svm.results
library(class)

pm.knn12<-knn(pm.tr.sc[,-8],pm.ts.sc[,-8],pm.tr.sc[,8],k=16)
(pm.kn.cm12<-table(pm.knn12,pm.ts.sc$Class))
list(cbind(sum(diag(pm.kn.cm12)/sum(pm.kn.cm12)),
pm.kn.cm12[4]/sum(pm.kn.cm12[,2]), #sensitivity
pm.kn.cm12[1]/sum(pm.kn.cm12[,1]))) #specificity

pm.knn12<-knn(pm.tr.sc[,c(1,2,5,6)],pm.ts.sc[,c(1,2,5,6)],pm.tr.sc[,8],k=7)
(pm.kn.cm12<-table(pm.knn12,pm.ts.sc$Class))
list(cbind(sum(diag(pm.kn.cm12)/sum(pm.kn.cm12)),
pm.kn.cm12[4]/sum(pm.kn.cm12[,2]), #sensitivity
pm.kn.cm12[1]/sum(pm.kn.cm12[,1]))) #specificity
# K accuracy optimizer
cv.error<-numeric(0)
t<-1
for (i in 1:200){

pm.knnK<-knn(pm.tr.sc[,c(1,2,5,6)],pm.ts.sc[,c(1,2,5,6)],pm.tr.sc[,8],k=t)
pm.kn.cmK<-table(pm.knnK,pm.ts.sc$Class)
cv.error[i]<-(sum(diag(pm.kn.cmK))/sum(pm.kn.cmK))

t<-t+1
}
cv.error<-matrix(cv.error)
cv.error.max<-max(cv.error)
(k.optm<-which.max(cv.error))
#K optimizer CV method
model1<-train(Class~glu+ped+npreg+bmi,data=pm.tr.sc,method='knn',
              tuneGrid=expand.grid(.k=1:10),
              metric='Accuracy',
              trControl=trainControl(
                method = 'repeatedcv',
                number = 10,
                repeats = 15))
model1$bestTune
set.seed(1234)
s.ptr <- data.frame(cbind(scale(Pim.tr[-8],center=TRUE)),Pim.tr[8])
s.pte <- data.frame(cbind(scale(Pim.te[-8],center=TRUE)),Pim.te[8])

invisible(rf <- randomForest(type ~.,data=s.ptr,ntree=1000,importance=TRUE))
pred <- predict(rf,newdata=s.pte[-8])
rf$confusion
```

```r
#cm for OOB
res.rfoob <- cbind(sum(diag(rf$confusion))/sum(rf$confusion),
rf$confusion[4]/sum(rf$confusion[,2]),
rf$confusion[1]/sum(rf$confusion[,1]))

colnames(res.rfoob)<-c('Accuracy','Sensitivity','Specificity');res.rfoob
#cm for test
(rf.cm <-table(s.pte[,8],pred))
res.rft <-cbind(sum(diag(rf.cm))/sum(rf.cm),
rf.cm[4]/sum(rf.cm[,2]), #sensitivity
rf.cm[1]/sum(rf.cm[,1])) #specificity

colnames(res.rft)<-c('Accuracy','Sensitivity','Specificity');res.rft

#variable importance
(imp_var <- data.frame(randomForest::importance(rf)))
#varImpPlot(rf)

#ggpairs(pima[c("npreg","glu","bp","skin","bmi","ped","age")])
pima <-rbind(Pima.tr,Pima.te)
pima <- pima[-c(204,296,417,79,292,175,132,72,320,520,428,490,399,210,403),]
train.ind = sample(1:517, 362, replace=FALSE)
test.ind = setdiff(1:517, train.ind)

pima.train <- pima[train.ind,]
pima.test <- pima[test.ind,]

rf.train.1 = pima.train[c("age","glu","ped","bmi")]
rf.label = as.factor(pima.train$type)

rf.1 = randomForest(x = rf.train.1, y = rf.label, importance = TRUE, ntree =2000)
rf.1.pred = predict(rf.1, pima.test)

cm = table(rf.1.pred,pima.test$type)
cm
sum(diag(cm))/sum(cm)
#cm_rfo <- confusionMatrix(rf.1.pred,pima.test$type)
#cm_rfo$table

#varImpPlot(rf.1)
#library(party)
x <- ctree(type ~ glu  + bmi + ped + age + npreg  , data=pima.train)
plot(x, type="simple")
pima.obj <- rfsrc(type~ npreg+glu+bp+skin+bmi+ped+age, data = pima, importance = TRUE)
find.interaction(pima.obj, method = "vimp", nrep=3)
find <-find.interaction(pima.obj)


pima <-rbind(Pima.tr,Pima.te)
pima <- pima[-c(204,296,417,79,292,175,132,72,320,520,428,490,399,210,403),]
train.ind = sample(1:517, 362, replace=FALSE)
test.ind = setdiff(1:517, train.ind)
pima.train <- pima[train.ind,]
pima.test <- pima[test.ind,]
fit1 = glm(type ~ glu  + bmi + ped + age , family = binomial(link = logit), data = pima.train)
#summary(fit1)
```

```
fit1$call
predicted <- ifelse(predict(fit1, newdata = pima.test, type = "response")<0.5, "No", "Yes")
(confusion <- table(predicted, factor(pima.test$type), dnn = c("Predicted type", "True type")))
list(cbind(sum(diag(confusion))/sum(confusion),
           confusion[4]/sum(confusion[,2]),#sensitivity
           confusion[1]/sum(confusion[,1])))#specificity

fit2 = glm(type ~ glu  + bmi + ped + age + bp , family = binomial(link = logit), data = pima.train)
#summary(fit2)
fit2$call
predicted2 <- ifelse(predict(fit2, newdata = pima.test, type = "response")<0.5, "No", "Yes")
(confusion2 <- table(predicted2, factor(pima.test$type), dnn = c("Predicted type", "True type")))
list(cbind(sum(diag(confusion2))/sum(confusion2),
           confusion2[4]/sum(confusion2[,2]),#sensitivity
           confusion2[1]/sum(confusion2[,1])))#specificity
anova(fit1,fit2,test="LRT")
train.control= trainControl(method="repeatedcv",repeats=5)
model <- train(type ~ ped*npreg + glu  + bmi + ped + age +npreg, data = pima, method = "glm",
               trControl = train.control)
#print(model)
par(mfrow=c(2,1))
g1.1 <- roc(type ~predict(fit1,newdata=pima.test,type='response'), data = pima.test)

g1.1 <- roc(type ~predict(fit1,newdata=pima.test,type='response'), data = pima.test)
plot.roc(g1.1, legacy.axes = TRUE )
auc(g1.1)

g1.2 <- roc(type ~predict(fit2,newdata=pima.test,type='response'), data = pima.test)

g1.2 <- roc(type ~predict(fit2,newdata=pima.test,type='response'), data = pima.test)
plot.roc(g1.2, legacy.axes = TRUE )
auc(g1.2)
resid.plot <- function(m.dat){
  ggplot(m.dat , aes( predictor.value, resid.predictor ) ) +
  geom_point(size = 0.3) +
  geom_smooth(method = "loess") +
  theme() +
  facet_wrap(~ predictor, scales = "free_x")
}

confusion.m <- function(m,data.tr){
  predicted <- ifelse(predict(m,newdata=data.tr,type='response')<.5,"No","Yes")
(confusion1.1 <- table(predicted,factor(data.tr$type),dnn=c("Predicted type","True type")))
}

diag.dat <- function(m, type){
  m.resid <- resid(m,type)
  data <- m.dat <- data.frame("resid.predictor" = as.vector(m.resid), "predictor" = rep(colnames(model.mat
  return(data)
}

partial_pearson <- function(fit)
{
  fit.dat <- diag.dat(fit,c("partial"))
  print(resid.plot(fit.dat))
  conf.q1 <- confusion.m(fit,pima.test)
  res<- residuals(fit, "pearson")
  print(summary(res))
}
par(mfrow=c(1,2))
partial_pearson(fit1)
partial_pearson(fit2)

fit1 = glm(type ~ glu  + bmi + ped + age , family = binomial(link = logit), data = pima.train)

summ <- summary(fit1)
summ$call
coeff<-data.frame(summ$coefficients)
(odds.inc <- round(exp(coeff$Estimate), 4) )
CI <- round(exp(confint(fit1,level = .95)),4)
CI
q5_1= glm(type ~ npreg , family = binomial(link = logit), data = pima.train)
s1 <-summary(q5_1)
list(s1$call,s1$aic)
q5_2 = glm(type ~ age, family = binomial(link = logit), data = pima.train)
s2 <-summary(q5_2)
list(s2$call,s2$aic)
q5_3 = glm(type ~npreg+age, family = binomial(link = logit), data = pima.train)
s3 <-summary(q5_3)
list(s3$call,s3$aic)
```