**STA 545 ASSIGNMENT #3**

**SURUCHI JAIKUMAR AHUJA**

**1 a)**

Here a model is developed to predict whether a given car gets high or low gas mileage based on the Auto data set in the ISLR package.

First a binary variable, mpg01 is created, that contains a 1 of mpg contains a value above its median, and a 0 if mph contains a value below its median. Using the data. frame () function a single data set is created containing both mpg01 and other Auto variables.

> mpg01 = rep(0, length(mpg))

> mpg01[mpg > median(mpg)] = 1

> mpg01

[1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 1 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1

[52] 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1

[103] 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0

[154] 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 0 1 0 1 1 0 1 1 1 1 1 1 1 0 0 0 0 0 0 1 0 1 1 1 1 0 0 0 0 1 1 1

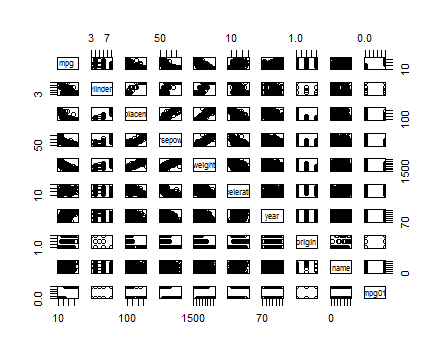
[205] 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 0 0 0 0 0 0 1 0

[256] 0 0 0 0 0 0 0 0 0 1 1 1 1 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

[307] 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

[358] 1 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1

The data is explored graphically in order to investigate the association between mph-1 and the other features.



The mp0g data is anti-correlated with cylinders, weight, displacement, horse power. Mpg, is a useful feature in helping predict mpg01.

**b)**

*Dividing the data set into Testing and Training set -*

The training set contains 207 observations of 10 variables and the test set contains 185 observations of 10 variables.

|  |
| --- |
| > dim(auto\_train)  [1] 207 10  > dim(auto\_test)  [1] 185 10 |

*Linear Discriminant Analysis*

The linear discriminant analysis is done using the “lda()” function from the R MASS package.

> lda.fit<-lda(mpg01~cylinders+displacement+horsepower+weight,data=auto\_train)

> lda.fit

Call:

lda(mpg01 ~ cylinders + displacement + horsepower + weight, data = auto\_train)

Prior probabilities of groups:

0 1

0.4782609 0.5217391

Group means:

cylinders displacement horsepower weight

0 6.787879 272.4848 125.94949 3612.121

1 4.203704 116.1296 78.27778 2344.769

Coefficients of linear discriminants:

LD1

cylinders -0.3605898446

displacement -0.0091952924

horsepower 0.0114624713

weight -0.0007326491

> lda\_train\_error

[1] 0.06280193

> lda\_test\_error

[1] 0.1405405

The Linear Discriminant Analysis error for the training set is 0.06280193 or 6.28% and for the test set is 0.1405405 or 14.05%.

*Quadratic Discriminant Analysis*

> qda.fit<-qda(mpg01~cylinders+displacement+horsepower+weight,data=auto\_train)

> qda.fit

Call:

qda(mpg01 ~ cylinders + displacement + horsepower + weight, data = auto\_train)

Prior probabilities of groups:

0 1

0.4782609 0.5217391

Group means:

cylinders displacement horsepower weight

0 6.787879 272.4848 125.94949 3612.121

1 4.203704 116.1296 78.27778 2344.769

> qda\_train\_error

[1] 0.0821256

> qda\_test\_error

[1] 0.1351351

The Quadratic Discriminant Analysis error for the training set is 0.0821256 or 8.21% and for the test set is 0.1351351 or 13.51%.

*K-Nearest Neighbors*

> knn.pred2 <- knn(auto\_train,auto\_test,auto\_train$mpg01,k=100)

>

> knn.pred2

[1] 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 1 0 0 1 0 1 1 1 1 0 0 0 0 0 0

[52] 0 0 0 0 0 0 1 1 1 1 1 0 0 1 1 0 1 0 0 1 1 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 0 1 1 0 1 0 1 1 1 1 0 0 0 0 1

[103] 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 1 1 1 0 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1

[154] 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 0 0 1 0 1 1 1

Levels: 0 1

>

> mean(knn.pred2 != mpg01.test)

[1] 0.1351351

The test error rate for k =100 is 0.1351351 or 13.51%.

*Logistic Regression*

> glm.fit <- glm(mpg01~cylinders+displacement+horsepower+weight,data=auto\_train,family="binomial")

> glm.fit

Call: glm(formula = mpg01 ~ cylinders + displacement + horsepower +

weight, family = "binomial", data = auto\_train)

Coefficients:

(Intercept) cylinders displacement horsepower weight

10.091627 0.064819 -0.026062 -0.024787 -0.001182

Degrees of Freedom: 206 Total (i.e. Null); 202 Residual

Null Deviance: 286.6

Residual Deviance: 99.57 AIC: 109.6

> train\_error\_lr

[1] 0.06763285

> test\_error\_lr

[1] 0.1459459

The Logistic Regression error for the training set is 0.06763285 or 6.76% and for the test set is 0.1459459 or 14.59%.

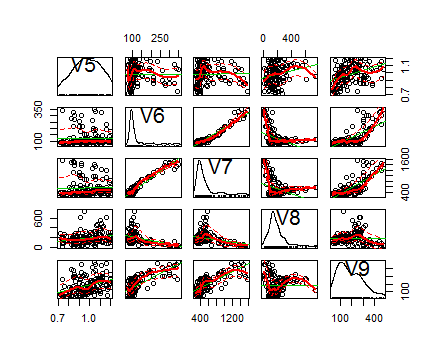
In terms of model interpretation, KNN shows a poor model interpretation, QDA has an average model interpretation, LDA is better than logistic regression in this case as there are 3 classes. Logistic Regression shows a good model interpretation only when there are less than 3 classes, so in this case it doesn’t interpret that well.

The results support the hypothesis regarding important predictors made in 1a, as the model interpreted in each method is accurate and the predictive accuracy is more in each case

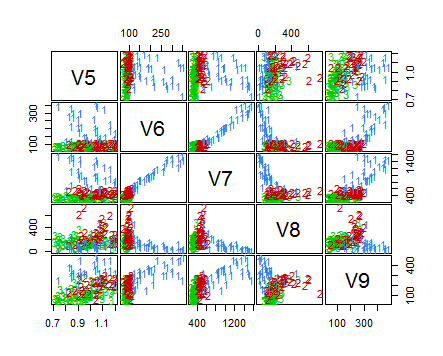
**2 a)**

The diabetes data set in the MMST package is used - Here the fourth column is the observation number, the 5-9 columns are variables – glucose.area, insulin.area, SSPG, relative.weight, and fasting.plasma.glucose. The final column is the class number.

*Scatterplot matrices*



*Pairwise Scatterplots*



In scatterplots, correlation is scale invariant. For a scatterplot of two variables the covariance measures ``how close the scatter is to a line''. Mathematical details follow but it should already be understood here that in this sense covariance measures only “linear dependence”. Here the classes have different covariance matrices and they are not multivariate normal. Looking at the pairwise scatterplots, the variables insulin.area and SSPG seem to be correlated.

**b)**

*Linear Discriminant Analysis –*

Also known as “canonical discriminant analysis”, or simply “discriminant analysis”. The purpose of linear discriminant analysis (LDA) is to find the linear combinations of the variables - glucose.area, insulin.area, SSPG, relative.weight, and fasting.plasma.glucose; that gives the best possible separation between the groups in the data set.

The linear discriminant analysis is done using the “lda()” function from the R MASS package.

|  |
| --- |
| dim(diabetes\_train)  [1] 95 6  > dim(diabetes\_test)  [1] 50 6 |

> lda.fit <- lda(V10~V5+V6+V7+V8+V9, data = diabetes\_train)

> lda.fit

Call:

lda(V10 ~ V5 + V6 + V7 + V8 + V9, data = diabetes\_train)

Prior probabilities of groups:

1 2 3

0.2736842 0.2210526 0.5052632

Group means:

V5 V6 V7 V8 V9

1 0.9726923 210.11538 1024.0769 110.8846 313.1923

2 1.0828571 98.28571 491.6667 294.8571 218.5238

3 0.9339583 91.77083 345.8333 173.8333 113.2500

Coefficients of linear discriminants:

LD1 LD2

V5 -1.4422552729 -4.349060975

V6 0.0408756164 0.028023089

V7 -0.0134224631 -0.005337782

V8 -0.0004299809 -0.006524894

V9 -0.0039300821 0.001407124

Proportion of trace:

LD1 LD2

0.8736 0.1264

The “proportion of trace” is the percentage separation achieved by each discriminant function. For example, for the diabetes data set we get the values as 87.36% and 12.64%.

> lda\_train\_error

[1] 0.08421053

> lda\_test\_error

[1] 0.14

The Linear Discriminant Analysis error for the training set is 0.08421053

Or 8.42% and for the test set is 0.14 or 14%.

*Quadratic discriminant analysis –*

It is closely related to [linear discriminant analysis](https://en.wikipedia.org/wiki/Linear_discriminant_analysis) (LDA), where it is assumed that the measurements from each class are [normally distributed](https://en.wikipedia.org/wiki/Normal_distribution).

> qda.fit <- qda(V10~V5+V6+V7+V8+V9, data = diabetes\_train)

> qda.fit

Call:

qda(V10 ~ V5 + V6 + V7 + V8 + V9, data = diabetes\_train)

Prior probabilities of groups:

1 2 3

0.2736842 0.2210526 0.5052632

Group means:

V5 V6 V7 V8 V9

1 0.9726923 210.11538 1024.0769 110.8846 313.1923

2 1.0828571 98.28571 491.6667 294.8571 218.5238

3 0.9339583 91.77083 345.8333 173.8333 113.2500

> qda\_train\_error

[1] 0.05263158

> qda\_test\_error

[1] 0.06

The Quadratic Discriminant Analysis error for the training set is 0.05263158

Or 5.26% and for the test set is 0.06 or 6%.

The performance of QDA is better than LDA in this case, since the QDA train error is much lesser than the LDA train error.

**c)**

Consider an individual having glucose area = 0.98, insulin area =122, SSPG = 544. Relative weight = 186, fasting plasma glucose = 184. Using the results of the above model and errors, it is seen that the Linear Discriminant Analysis assigns this individual to class 2 and Quadratic Discriminant Analysis assigns this individual to class 1.

``````````

**3 a)**

Under the assumptions in the logistic Regression model, proving that sum of posterior probabilities of k classes equal to one

Considering k classes,

Posterior probabilities are :-

P ( G = 1 | X = x) =

P ( G = 2 | X = x) =

.

.

.

.

.

.

P ( G = (k-1) | X = x) =

P ( G = k | X = x) =

Sum of posterior Probabilities for k classes is :-

+ ………. + +

**b)**

Here we use simple algebra to show that the logistic function representation and the logit representation for the logistic regression model are equivalent.

So we have p(x) =

Now we have to solve for

**4 a)**

*Leave-one-out cross-validation (LOOCV)*

It involves splitting the set of observations into two parts. A single observation (x1,y1) is used for the validation set, and the remaining observations {(x2,y2),...,(xn,yn)} make up the training set.

Computing the LOOCV errors from fitting the following four models using least squares:

Y = + X + ε

Y = + X + + ε

Y = + X + + + ε

Y = + X + + + + ε

> loocverror

[[1]]

[1] 7.288162

[[2]]

[1] 0.9374236

[[3]]

[1] 0.9566218

[[4]]

[1] 0.9539049

So the first model has a loocv error of 7.288162, the second model has a loocv error of 0.9374236, the third model has a loocv error of 0.9566218, the fourth model has a loocv error of 0.9539049.

By comparing the loocv errors of all the four models the model 2 seems to have the smallest prediction error of 0.937.

Yes, it was expected and we tend to use the model with X and it is more accurate with a smaller prediction error.

**b)**

Output :

> summary(glmfiti)

Call:

glm(formula = y ~ poly(x, 4), data = dat)

Deviance Residuals:

Min 1Q Median 3Q Max

-2.0550 -0.6212 -0.1567 0.5952 2.2267

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -1.55002 0.09591 -16.162 < 2e-16 \*\*\*

poly(x, 4)1 6.18883 0.95905 6.453 4.59e-09 \*\*\*

poly(x, 4)2 -23.94830 0.95905 -24.971 < 2e-16 \*\*\*

poly(x, 4)3 0.26411 0.95905 0.275 0.784

poly(x, 4)4 1.25710 0.95905 1.311 0.193

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for gaussian family taken to be 0.9197797)

Null deviance: 700.852 on 99 degrees of freedom

Residual deviance: 87.379 on 95 degrees of freedom

AIC: 282.3

Number of Fisher Scoring iterations: 2

The coefficient standard error for the four models are the same. The coefficient estimate for the second model is a negative value, whereas for the other three there are positive values obtained. The coefficients are statistically significant, as all the p values obtained in the summary have a value lesser than 0.05.

These results agree with the conclusions drawn from the cross validation.

-------------------------------------------------------------------------------------------------------------------------------------------- The End ----------------------------------------------------------------------------------------------------------------------------------------------------