STATS 790
Assignment 1

Doudou Jin – 400174871

1/20/23

Q1

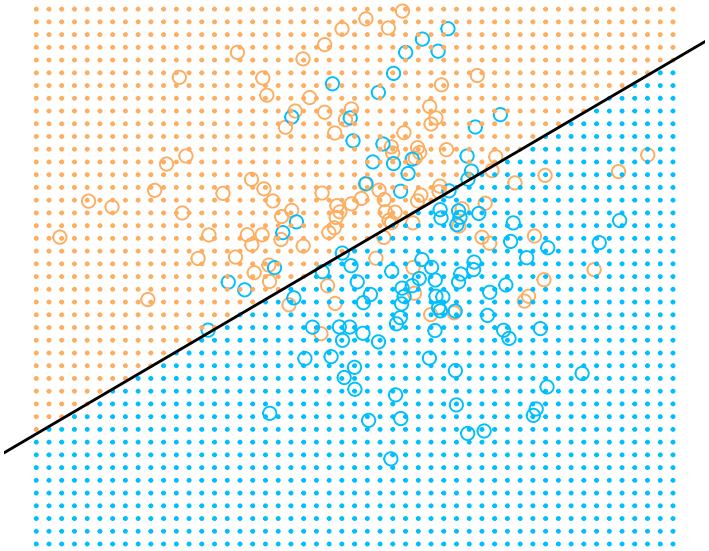
I agree with Breiman (Breiman 2001)'s remarks that Statisticians shouldn't restrict their problem-solving potentials by the hard rules of traditional data models such as the goodness-of-fit and residual analysis. Data from our modern world is coming in with hundreds and thousands of features, it would be a good idea for the Statisticians to work with the Computer Scientists, so that they can develop efficient algorithmic solutions while validating statistical theories and assumptions against the data and the solutions. This means that with expertise from both subjects, we can strive to realize the full potential of accuracy and interpretability.

Q2

```r
library(ggplot2)

set.seed(1)
#orange
x1_A <- rnorm(100, mean = 0.4, sd = 0.5)
x2_A <- rnorm(100, mean = 0.7, sd = 0.3)
y_A <- rep(1, 100)
group_A <- data.frame(x1_A, x2_A, y_A)
colnames(group_A) = c("x1", "x2", "y")

set.seed(1)
#blue
x1_B <- rnorm(100, mean = 0.65, sd = 0.35)
x2_B <- rnorm(100, mean = 0.4, sd = 0.4)
y_B <- rep(0, 100)
group_B <- data.frame(x1_B, x2_B, y_B)
colnames(group_B) = c("x1", "x2", "y")

#training set
q2 <- rbind(group_A, group_B)

#linear regression model
linear_reg <- lm(y~x1+x2, data=q2)
# y = 0.3895 - 0.2774 x1 + 0.4779 x2

#test data with decision boundary 0.5
x1_bg <- seq(-0.8, 1.7, 0.05)
x2_bg <- seq(-0.7, 1.4, 0.05)
bg <- expand.grid(x1_bg, x2_bg)
colnames(bg) <- c("x1", "x2")
```

```r
y_pred <- predict(linear_reg, newdata = bg)
y_pred <- ifelse(y_pred < 0.5, 0, 1)
bg_pred <- data.frame(bg, y_pred)

#plot
ggplot() +
  geom_point(data = q2
             , aes(x=x1, y=x2, color = factor(y))
             , size = 2
             , shape=1
             ) +
  geom_point(data = bg_pred
             , aes(x=x1, y=x2, color = factor(y_pred))
             , size = 0.2
             ) +
  geom_abline(intercept = (0.5-coef(linear_reg)[1])/coef(linear_reg)[3]
             , slope = -coef(linear_reg)[2]/coef(linear_reg)[3]
             ) +
  scale_color_manual(values = c("#00BFFF", "#FDAE61")
                     , guide = "none"
                     ) +
  theme(axis.text.x=element_blank()
        , axis.title.x=element_blank()
        , axis.ticks.x=element_blank()
        , axis.text.y=element_blank()
        , axis.title.y=element_blank()
        , axis.ticks.y=element_blank()
        , panel.background = element_rect(fill = "white")
        ) +
  ggtitle("Linear Regression of 0/1 Response") +
  coord_fixed()
```

Linear Regression of 0/1 Response



Q3

$$\begin{aligned} MAE(m) &= E[|Y-m|] \\ &= \int_{-\infty}^{\infty} |y-m| p(y) dy \\ &= \int_{-\infty}^{m} (m-y) p(y) dy + \int_{m}^{\infty} (y-m) p(y) dy \end{aligned}$$

Find the derivative of MAE with respect to m, then

$$\frac{d}{dm} MAE(m) = \frac{d}{dm} \int_{-\infty}^{m} (m-y) p(y) dy + \frac{d}{dm} \int_{m}^{\infty} (y-m) p(y) dy$$

By using Leibniz Integral Rule,

$$\begin{aligned} \frac{d}{dm} \int_{-\infty}^{m} (m-y) p(y) dy &= \frac{d}{dm}(m)(m-m)p(y) + \frac{d}{dm}(-\infty)(m+\infty)p(y) + \int_{-\infty}^{m} \frac{\partial}{\partial m}(m-y)p(y)dy \\ &= 0 + 0 + \int_{-\infty}^{m} \frac{\partial}{\partial m}(m-y)p(y)dy \\ &= \int_{-\infty}^{m} \frac{\partial}{\partial m} mp(y) - yp(y)dy \\ &= \int_{-\infty}^{m} p(y)dy \end{aligned}$$

$$\begin{aligned} \frac{d}{dm} \int_{m}^{\infty} (y-m) p(y) dy &= \frac{d}{dm}(m)(m-m)p(y) + \frac{d}{dm}(\infty)(\infty-m)p(y) + \int_{m}^{\infty} \frac{\partial}{\partial m}(y-m)p(y)dy \\ &= 0 + 0 + \int_{m}^{\infty} \frac{\partial}{\partial m}(y-m)p(y)dy \\ &= \int_{m}^{\infty} \frac{\partial}{\partial m} yp(y) - mp(y)dy \\ &= -\int_{m}^{\infty} p(y)dy \end{aligned}$$

Thus,

$$MAE(m) = \int_{-\infty}^{m} p(y)dy - \int_{m}^{\infty} p(y)dy$$

To minimize the MAE, set it to zero,

$$\frac{d}{dm} MAE(m)\Big|_{m=\tilde{\mu}} = 0$$

Then,

$$\begin{aligned}
\int_{-\infty}^{\tilde{\mu}} p(y)dy - \int_{\tilde{\mu}}^{\infty} p(y)dy &= 0 \\
\Rightarrow \int_{-\infty}^{\tilde{\mu}} p(y)dy &= \int_{\tilde{\mu}}^{\infty} p(y)dy \\
\Rightarrow P(Y \leq \tilde{\mu}) &= P(Y > \tilde{\mu})
\end{aligned}$$

Since $P(Y \leq \tilde{\mu}) + P(Y > \tilde{\mu}) = 1$,

$$P(Y \leq \tilde{\mu}) = P(Y > \tilde{\mu}) = \frac{1}{2}$$

By the definition of median, $\tilde{\mu}$ is the median of Y. Thereforem when $\tilde{\mu}$ is the median, the MAE is minimized.

$MSE(m) = E[(Y - m)^2]$ is good at highlighting the large errors. Since MSE is minimized by the mean and the MAE is minimized by the median, MSE could give more weights to those observations that are far away from the mean, and punish large errors in prediction, however, with MAE, the fitted model would be closer to the median and may lead to bias. Thus, in general, MSE is more useful as a measure of error than MAE.

Q4

When global mean as a linear smoother, the diagonal of influence matrix $\widehat{w}$ will be $\frac{1}{n}$, and by equation 1.70 (Shalizi 2013),

$$tr(w) = \sum_{i=1}^{n} w(x_i, x_i) = n * \frac{1}{n} = 1$$

Therefore, when global mean as a linear smoother, it has 1 degree of freedom.

Q5

When k-nearest-neighbors regression as a linear smoother, by equation 1.55 (Shalizi 2013),

$$\widehat{w}(x_i, x) = \begin{cases} \frac{1}{k} & x_i \text{ one of the k nearest neighbors of } x \\ 0 & \text{otherwise} \end{cases}$$

Therefore, the diagonal of of influence matrix $\widehat{w}$ is always $\frac{1}{k}$, since x_i is always one of the nearest neighbors of itself. By equation 1.70 (Shalizi 2013), the degree of freedom of the linear smoother is

$$tr(w) = \sum_{i=1}^{n} w(x_i, x_i) = n * \frac{1}{k} = \frac{n}{k}$$

So, when k-nearest-neighbors regression as a linear smoother, it has $\frac{n}{k}$ degree of freedoms, and when $k = n$, it has one degree of freedom.

Q6

```
#filter()
library(dplyr)
#knn
library(class)
#classAgreement
library(e1071)
#kable
library(knitr)

zip_train <- read.table("zip.train", quote="\"", comment.char="")
zip_test <- read.table("zip.test", quote="\"", comment.char="")

#2's and 3's
zip_train <- zip_train %>% filter(V1 == "2" | V1 == "3")
zip_test <- zip_test %>% filter(V1 == "2" | V1 == "3")

#linear regression
q6_linear_reg <- lm(V1~., data = zip_train)

#fit training set
train_fit <- predict(q6_linear_reg, newdata = zip_train)
train_fit <- ifelse(train_fit < 2.5, 2, 3)
train_fit <- factor(train_fit)
tab <- table(train_fit, factor(zip_train[,1]))
1-classAgreement(tab)$diag

#fit test set
test_fit <- predict(q6_linear_reg, newdata = zip_test)
test_fit <- ifelse(test_fit < 2.5, 2, 3)
test_fit <- factor(test_fit)
```

```r
tab2 <- table(test_fit, factor(zip_test[,1]))
1-classAgreement(tab2)$diag

#knn, k=1, fit training set
set.seed(1)
q6_knn1 <- knn(zip_train[,-1], zip_train[,-1], cl = zip_train[,1], k=1)
tab_knn1 <- table(zip_train[,1], q6_knn1)
1-classAgreement(tab_knn1)$diag

#knn, k=3, fit training set
set.seed(1)
q6_knn3 <- knn(zip_train[,-1], zip_train[,-1], cl = zip_train[,1], k=3)
tab_knn3 <- table(zip_train[,1], q6_knn3)
1-classAgreement(tab_knn3)$diag

#knn, k=5, fit training set
set.seed(1)
q6_knn5 <- knn(zip_train[,-1], zip_train[,-1], cl = zip_train[,1], k=5)
tab_knn5 <- table(zip_train[,1], q6_knn5)
1-classAgreement(tab_knn5)$diag

#knn, k=7, fit training set
set.seed(1)
q6_knn7 <- knn(zip_train[,-1], zip_train[,-1], cl = zip_train[,1], k=7)
tab_knn7 <- table(zip_train[,1], q6_knn7)
1-classAgreement(tab_knn7)$diag

#knn, k=15, fit training set
set.seed(1)
q6_knn15 <- knn(zip_train[,-1], zip_train[,-1], cl = zip_train[,1], k=15)
tab_knn15 <- table(zip_train[,1], q6_knn15)
```

```r
1-classAgreement(tab_knn15)$diag

#knn, k=1, fit test set
set.seed(1)
q6_knn1_t <- knn(zip_train[,-1], zip_test[,-1], cl = zip_train[,1], k=1)
tab_knn1_t <- table(zip_test[,1], q6_knn1_t)
1-classAgreement(tab_knn1_t)$diag

#knn, k=3, fit test set
set.seed(1)
q6_knn3_t <- knn(zip_train[,-1], zip_test[,-1], cl = zip_train[,1], k=3)
tab_knn3_t <- table(zip_test[,1], q6_knn3_t)
1-classAgreement(tab_knn3_t)$diag

#knn, k=5, fit test set
set.seed(1)
q6_knn5_t <- knn(zip_train[,-1], zip_test[,-1], cl = zip_train[,1], k=5)
tab_knn5_t <- table(zip_test[,1], q6_knn5_t)
1-classAgreement(tab_knn5_t)$diag

#knn, k=7, fit test set
set.seed(1)
q6_knn7_t <- knn(zip_train[,-1], zip_test[,-1], cl = zip_train[,1], k=7)
tab_knn7_t <- table(zip_test[,1], q6_knn7_t)
1-classAgreement(tab_knn7_t)$diag

#knn, k=15, fit test set
set.seed(1)
q6_knn15_t <- knn(zip_train[,-1], zip_test[,-1], cl = zip_train[,1], k=15)
tab_knn15_t <- table(zip_test[,1], q6_knn15_t)
1-classAgreement(tab_knn15_t)$diag
```

```r
method <- c("Linear Regression", "KNN, k=1", "KNN, k=3", "KNN, k=5",
            "KNN, k=7", "KNN, k=15")
training_error <- c(0.005759539, 0, 0.005039597, 0.005759539, 0.006479482,
            0.009359251)
test_error <- c(0.04120879, 0.02472527, 0.03021978, 0.03021978,
            0.03296703, 0.03846154)
cp <- data.frame(method, training_error, test_error)
kable(cp, caption = "Classification Performance")
```

Table 1: Classification Performance

method	training_error	test_error
Linear Regression	0.0057595	0.0412088
KNN, k=1	0.0000000	0.0247253
KNN, k=3	0.0050396	0.0302198
KNN, k=5	0.0057595	0.0302198
KNN, k=7	0.0064795	0.0329670
KNN, k=15	0.0093593	0.0384615

In conclusion, the training errors are higher than test errors for all methods, indicating that the model is overfitting to the training set. Additionally, knn performs better than linear regression. All choices of k have lower test error rate than linear regression. The best performance is achieved when using K=1, as it has the lowest test error rate among all the methods.

Reference

Breiman, Leo. 2001. "Statistical Modeling: The Two Cultures." *Statistical Science* 16 (3): 199–215. http://www.jstor.org/stable/2676681.

Meyer, David, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, and Friedrich Leisch. 2022. *E1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. https://CRAN.R-project.org/package=e1071.

Miller, Andrew C., Nicholas J. Foti, and Emily B. Fox. 2021. "Breiman's Two Cultures: You Don't Have to Choose Sides." https://doi.org/10.48550/ARXIV.2104.12219.

Raper, Simon. 2020. "Leo Breiman's 'Two Cultures'." *Significance* 17 (1): 34–37. https://doi.org/10.1111/j.1740-9713.2020.01357.x.

Shalizi, Cosma. 2013. "Advanced Data Analysis from an Elementary Point of View."

Tibshirani, R., T. Hastie, and J. H. Friedman. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction : With 200 Full-Color Illustrations*. Springer Series in Statistics. Springer. https://books.google.ca/books?id=SECjnQAACAAJ.

Venables, W. N., and B. D. Ripley. 2002. *Modern Applied Statistics with s*. Fourth. New York: Springer. https://www.stats.ox.ac.uk/pub/MASS4/.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2022. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Xie, Yihui. 2014. "Knitr: A Comprehensive Tool for Reproducible Research in r." In *Implementing Reproducible Computational Research*, edited by Victoria Stodden, Friedrich Leisch, and Roger D. Peng. Chapman; Hall/CRC. http://www.crcpress.com/product/isbn/9781466561595.