ACSC/STAT 3740, Predictive Analytics

WINTER 2024

Toby Kenney

Homework Sheet 4

Model Solutions

Note: All data sets in this homework are simulated.

Standard Questions

1. *The file* `HW4Q1.txt` *contains data on the relation between workers' rights and happiness. The data set contains the following variables:*

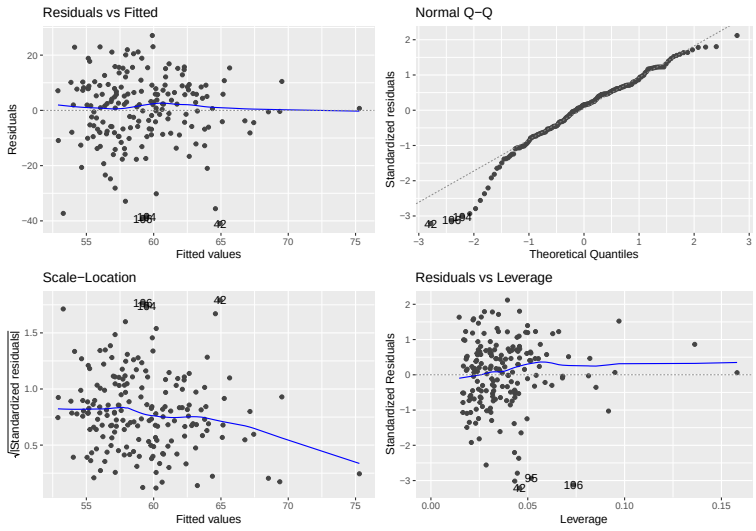
Variable	*Meaning*
max.weekly.hours	*The maximum number of hours an employee can be regularly required to work in*
min.hourly.wage	*The minimum hourly wage that can be paid to an employee.*
paid.sick.leave	*Whether employees are legally entitled to paid sick leave.*
paid.parental.leave	*Whether employees are legally entitled to paid parental leave.*
min.holidays	*The minimum number of holidays that employees are entitled to.*
union.percent	*The percentage of employees who belong to a labour union.*
happiness	*An index indicating the overall happiness of the population.*

A data analyst uses the following code to fit a linear regression model to the data.

```
HW4Q1_linear<-lm(happiness~., data=HW4Q1)
```

Use appropriate diagnostics to assess how appropriate the assumptions of the linear regression model are. What changes would you suggest making to the model to better model the data?

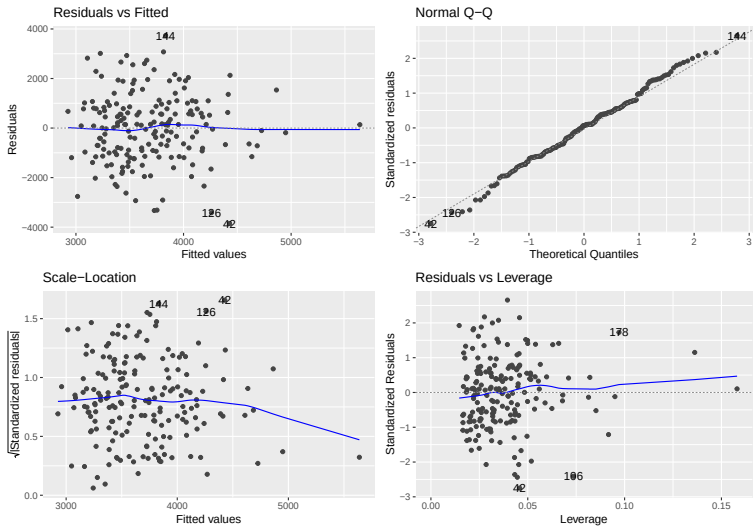
We first plot the standard diagnostics:



From the plot of fitted values against residuals, we do not see evidence for non-linear effects. As the distribution of fitted values is slightly heavy-tailed, it is unclear whether there is heteroskedasticity. The scale-location plot indicates there may be slightly higher variance for small fitted values. From the Q-Q plot, the lower-tail of the residuals is clearly heavier than a normal distribution, suggesting that the distribution of the residuals is skewed. Looking at the residuals vs. leverage plot, there are some points with large leverage, but the residuals of these points are relatively small, suggesting that while these points are influential, they are consistent with other data points.

A natural first adjustment to the model is to change the response, either via a GLM or by transforming the response. Since the response is in the interval $[0, 100]$, it makes sense that a normal distribution is inappropriate. We could consider a logistic transformation $\tilde{x} = \log\left(\frac{x}{100-x}\right)$. However, this produces very similar diagnostic plots.

Examining the large negative residuals, there is no clear pattern to them, except that they correspond to the smallest values of happiness. This suggests that a suitable non-linear transformation of happiness could fix this issue, but it is not completely clear what transformation would work best. We try squaring the happiness variable, and get the following diagnostic plots:



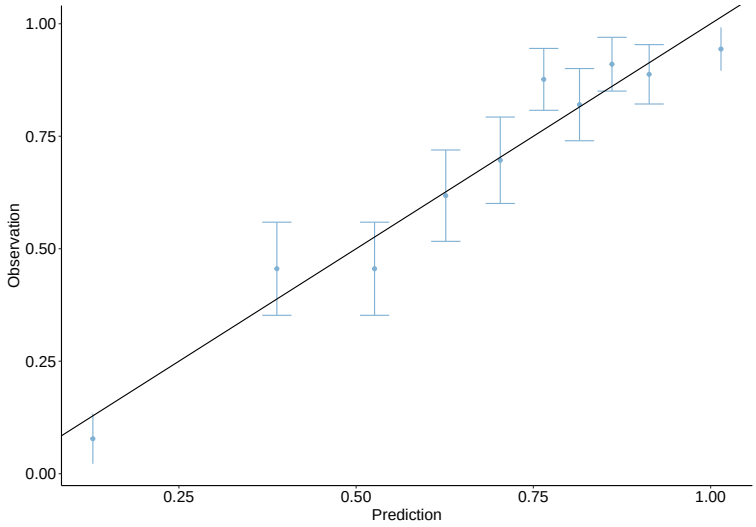
These suggest that the linear model is appropriate. There is one high-leverage point, which we might consider removing to check its influence.

2. *A data scientist at a company is analysing data about customer retention in the file* `HW4Q2.txt`.

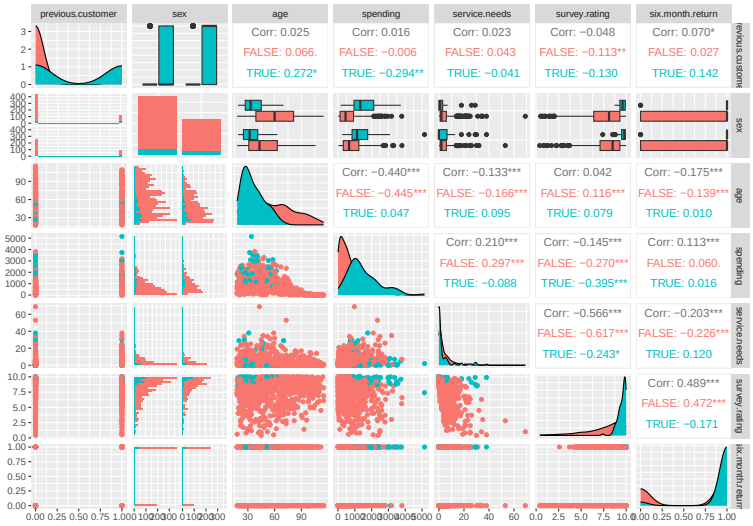
Variable	*Meaning*
previous.customer	*Whether the customer has previously done business with the company.*
age	*The customer's age*
sex	*The customer's gender*
spending	*How much the customer spent.*
service.needs	*The number of hours of service the customer needed*
survey.rating	*The rating given by the customer.*
six.month.return	*Whether the customer returned within six months.*

She has fitted a generalised linear model to predict whether the customer returns within 6 months, using the code in the file `HW4Q2_GLM.R`. *Perform diagnostics to test which of the assumptions of this model are reasonable. What changes would you suggest making to the model to better model the data?*

We start with a calibration plot, to see how the estimated probabilities correspond to reality.



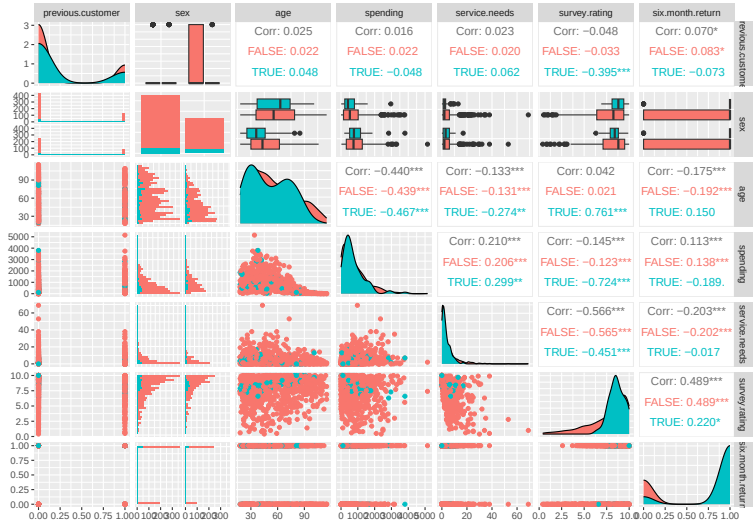
We see that the probabilities are reasonably calibrated, with the exception of a few points, where the observed probability is significantly different from the observed probability. One of these points is where the predicted probability is close to 1. This could be caused by some of the numerical variables being heavy-tailed, in which case it might be fixed by transforming those predictors. We can look at the points where we overestimate the probability:



We see that these are mostly the points with very high survey rating and fairly high spending. This suggests that a non-linear transformation of survey rating or spending could improve the model.

We can also look at the other group where the model underestmates the

probability.



This seems to have fairly high survey rating and lower spending, suggesting that adding an interaction between survey rating and spending might improve the model.

3. *A scientist is reviewing data about the factors affecting health of captive animals, in the file* `HW4Q3.txt`.

Variable	*Meaning*
social.type	*The type of social group that the animal usually lives in in the wild.*
diet	*The animal's diet — herbivore, carnivore, etc.*
born	*Whether the animal was born in captivity.*
enclosure.size	*The size of the enclosure in which the animal is kept.*
body.weight	*The animal's body weight.*
enclosure.shared	*The number of other animals sharing the enclosure.*
health.index	*An overall assessment of the animal's health.*

He has fitted a generalised additive model, a random forest model and a generalised linear model including a number of interaction terms and polynomial terms, to predict the health index, using the code in the file `HW4Q3_models.R`. *Assess which of these models is better at predicting the data. [You may need to modify the code provided to do this.]*

We use 10-fold cross-validation

```
HW4Q3<-read.table("HW4Q3.txt")

library(mgcv)
library(caret)
library(dplyr)

### Use cross-validation

nfold<-10

folds<-createDataPartition(HW4Q3$health.index, nfold)

MSE<-matrix(0,nfold,3)

GLM.Formula<-health.index~social.type+diet+born+enclosure.size+
                    body.weight+enclosure.shared+enclosure.size/sqrt(body.weight)+enclosure.size/(sqrt(enc

for(i in seq_len(nfold)){

     train.data<-HW4Q3[-folds[[i]],]
     test.data<-HW4Q3[folds[[i]],]

     GAM.Model<-gam(health.index~social.type+diet+born+s(enclosure.size)+
                         s(body.weight)+s(enclosure.shared),
                    data=train.data)

     GAM.pred<-predict(GAM.Model,newdata=test.data)
     MSE[i,1]<-sum((GAM.pred-test.data$health.index)^2)


     RF.Model<-train(train.data[,-7],
                     train.data[,7],
                     method="rf",
                     trControl=trainControl(method="repeatedcv",number=10,repeats=2),
                     tuneGrid=expand.grid(mtry=seq_len(5)),ntree=500)

     RF.pred<-predict(RF.Model,newdata=test.data)
     MSE[i,2]<-sum((RF.pred-test.data$health.index)^2)



     GLM.Model<-lm(GLM.Formula,data=train.data)
     GLM.pred<-predict(GLM.Model,newdata=test.data)
     MSE[i,3]<-sum((GLM.pred-test.data$health.index)^2)

}

colSums(MSE)/dim(HW4Q3)[1]
```

This calculates the following MSE:

Method	MSE
GAM	1317.0100
Random Forest	883.1466
GLM	1276.0263

Thus, random forest does best at predicting the health index.

4. *The file* `HW4Q4.txt` *contains data from about the probability that an individual will be injured during a sports match. The data set contains the following variables:*

Variable	*Meaning*
age	*The age of the participant.*
sex	*The sex of the participant.*
contact	*Whether the sport is a contact sport.*
match.length	*The length of the match.*
fitness	*An overall assessment of the fitness level of the individual.*
strength	*A measure of the strength of the individual.*
previous.injury	*Whether the individual has been injured in the previous six months.*
injured	*Whether the individual is injured.*

A data analyst uses the following code to fit a decision tree to the data:

```
HW4Q4<-read.table("HW4Q4.txt")

library(rpart)

HW4Q4_dt<-rpart(formula=injured ~ .,
                data=HW4Q4,
                control=rpart.control(minbucket=1, # smallest size of node
                                      maxdepth=10, # largest depth of tree.
                                      cp=0.000001)) # complexity

### Find the minimum cross-validated error.
### Using 1-s.e. chooses a very simple tree.
HW4Q4_min<-min(HW4Q4_dt$cptable[,4])
HW4Q4_which_min<-min(which(HW4Q4_dt$cptable[,4]==HW4Q4_min))
HW4Q4_cp_min<-HW4Q4_dt$cptable[HW4Q4_which_min,1]
HW4Q4_dt_min<-prune(HW4Q4_dt, cp=HW4Q4_cp_min)
```

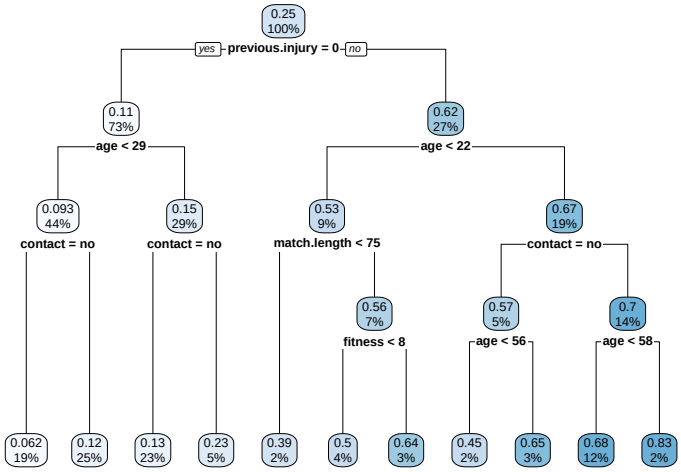
and uses the following code to select variables using stepwise regression with AIC:

```
HW4Q4_Null_model<-glm(injured ~1, data=HW4Q4, family=binomial(link=logit))

HW4Q4_Full_model<-glm(injured ~., data=HW4Q4, family=binomial(link=logit))


library(MASS)
HW4Q4_Stepwise<-stepAIC(HW4Q4_Null_model,
                        direction="both",
                        scope=list(lower=HW4Q4_Null_model,
                                   upper=HW4Q4_Full_model))
```

The code is in the files `HW4Q4_Decision_tree.R` *and* `HW4Q4_Stepwise_AIC.R` *respectively.*

Based on the results of these analyses, should she try to adjust the models to better fit the data, and if so, how might she do so?

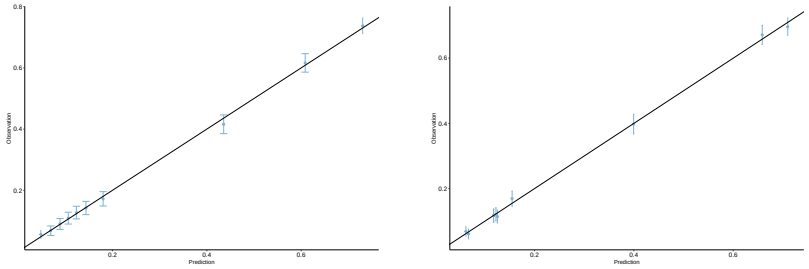
We plot the fitted decision tree.



This shows some possible interactions, which could be added to the model.

The Stepwise AIC selects the variables `previous.injury`, `age`, `contact`, `match.length`, `fitness` and `strength`. The first five of these are in the decision tree.

We next look at calibration plots:



Both models seem well calibrated.

From the decision tree, there could be interaction terms between some of the predictors. In particular, `age` and `contact` appear often in the tree, so adding an interaction term between these may improve prediction.

We may also compare prediction performance for the decision tree and random forest. We see that the log-likelihood loss for decision tree is 0.7139728, while for random forest, it is 0.4670339. For the stepwise

method, the deviance is 8464 from 9983 observations. This means that the negative log-likelihood is 4232, so the average negative log-likelihood per observation is $4232/9983 = 0.423920665131$. This is on training data, and based on variable selection, so is not perfectly comparable with random forest and decision tree.

This indicates that the decision tree is oversimplified, and that interactions are not significant. Overall few changes if any are needed to the stepwise model.

These analyses and plots used the following code:

```
library(rpart.plot)
rpart.plot(HW4Q4_dt_min)


summary(HW4Q4_Stepwise)


library(predtools)

calibration_plot(data.frame("pred"=predict(HW4Q4_Stepwise,type="response"),
                            "true"=HW4Q4$injured),pred="pred",obs="true")$
    calibration_plot


calibration_plot(data.frame("pred"=predict(HW4Q4_dt_min,type="vector"),
                            "true"=HW4Q4$injured),pred="pred",obs="true")$
    calibration_plot




HW4Q4_RF<-train(plyr::revalue(as.factor(injured),c("0"="no","1"="yes"))~.,
                data=HW4Q4,
                method="rf",
                trControl=trainControl(method="repeatedcv",
                                       number=10,
                                       repeats=2,
                                       classProbs=TRUE,
                                       summaryFunction=mnLogLoss),
                tuneGrid=expand.grid(mtry=seq_len(7)),
                ntree=500)

### Compare results for different methods.
HW4Q4_RF$results
HW4Q4_dt_min$cptable
HW4Q4_Stepwise
```