DATA303 Test 2

NAME:Nicholas Tran ID:300296259

10 June 2022

```
# Preprocessing data set
library(ISLR)
library(glmnet)
data(Hitters)
Hitters <- Hitters[, -c(14, 15, 20)]
Hitters <- na.omit(Hitters)
set.seed(11)
```

1. (55 marks)

a. **(5 marks)**

```
train_index <- sample(nrow(Hitters), nrow(Hitters)*0.7)
train = Hitters[train_index,]
test = Hitters[-train_index,]
dim(train)
```

```
## [1] 184  17
```

```
dim(test)
```

```
## [1] 79 17
```

b. **(10 marks)**

```
fit1 <- lm(Salary ~ ., data = train)

y <- test$Salary
y_hat <- predict(fit1, newdata = test)

MSE <- mean((y-y_hat)^2)
MSE
```

```
## [1] 152555.7
```

c. **(15 marks)**

```
y <- train$Salary
x <- model.matrix(Salary ~., train)
y.test <- test$Salary
x.test <- model.matrix(Salary ~., test)

lasso.mod <- glmnet(x, y, alpha=1)
grid <- 10 ^ seq(-3, 1, length = 5)
cv.out <- cv.glmnet(x, y, alpha=1, lambda = grid)
```

```
lam1se <- cv.out$lambda.1se
lam1se
```

```
## [1] 10
```

```
log(lam1se)
```

```
## [1] 2.302585
```

```
lasso.predict <- predict(lasso.mod, s=lam1se, newx = x.test)
MSE <- mean((lasso.predict - y.test)^2)
MSE
```

```
## [1] 165021.8
```

```
lasso.coeff <- predict(lasso.mod, type="coefficients", s=lam1se)
lasso.coeff
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept) -158.0886715
## (Intercept)    .
## AtBat          .
## Hits           2.4166043
## HmRun          .
## Runs           .
## RBI            .
## Walks          3.2649464
## Years          .
## CAtBat         .
## CHits          .
## CHmRun         0.8109273
## CRuns          0.2160730
## CRBI           0.3506870
## CWalks         .
## PutOuts        0.2338619
## Assists        .
## Errors        -0.8692191
```

d. **(17 marks)**

```
fit2 <- lm(Salary ~ Hits + HmRun + Runs + Walks + CRuns + CRBI + PutOuts + Errors, data = Hitters)
summary(fit2)
```

```
##
## Call:
## lm(formula = Salary ~ Hits + HmRun + Runs + Walks + CRuns + CRBI +
##     PutOuts + Errors, data = Hitters)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -928.01 -167.21  -39.13  110.61 2201.89
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -84.26213   58.56314  -1.439 0.151431
## Hits          2.49994    1.22541   2.040 0.042377 *
## HmRun        -1.28453    3.64594  -0.352 0.724893
```

```
## Runs         -0.33527    2.65658  -0.126 0.899670
## Walks         2.62500    1.42296   1.845 0.066239 .
## CRuns         0.24278    0.21744   1.117 0.265239
## CRBI          0.41250    0.23151   1.782 0.075981 .
## PutOuts       0.26755    0.07942   3.369 0.000872 ***
## Errors       -3.00220    3.37323  -0.890 0.374304
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 334.4 on 254 degrees of freedom
## Multiple R-squared:  0.4673, Adjusted R-squared:  0.4506
## F-statistic: 27.86 on 8 and 254 DF,  p-value: < 2.2e-16
```

Hits and PutOuts are significant predictors of Salary as their p-values are below the standard confidence threshold of $\alpha = 0.05$.

However HmRun, Runs, Walks, CRuns, CRBI and Errors are not significant predictors of Salary.

It's not the best model since there are many predictors that are not significant predictors of Salary, best subset selection should be applied to ascertain which predictors should be kept or removed.

e. **(10 marks)**

Using the train MSE only can't give you information on whether or not the model generalises well and you can't detect overfitting/underfitting. Test MSE is a useful metric as it allows your model to test it's performance on unseen data.

2. (30 marks)

a. **(10 marks)**

$H_0 : \beta_1 = 0$

$H_1 : \beta_1 \neq 0$

Test Statistic: z-Statistic = 1.324

p-value = 0.1855

Conclusion: The p-value is larger than any reasonable confidence interval so there's insufficient evidence to reject the null hypothesis, thus there's no to little evidence that β_1 deviates from 0, and there's no evidence to suggest that there is a statistically significant relationship between FIRE and Temperature.

b. **(10 marks)**

$H_0 : \beta_2 = \beta_3 = \beta_4 = 0$

H_1 : At least one $\neq 0$

Test Statistic: z-Statistics = 0.3948, 0.3544 and 0.4672 (β_2, β_3, β_4) respectfully.

p-values = 0.2638, 0.08575 and 0.009038 (β_2, β_3, β_4) respectfully.

Conclusion:

All of the Seasons' p-values are not significantly low besides from Winter, so only β_4 is significantly different from 0 which means that only the Winter season has a significant relationship with occurrences of forest fires when other predictors are kept constant.

c. **(5 marks)**

The odds of of Fire occurring when the season is Winter is estimated to be 1.22 that compared to the odds of a Fire occurring when the season is Spring. given that all other predictors are kept constant. (The odds of a Fire occurring in Winter is estimated to be 1.22 times the odds of a Fire occurring when it's Spring)

d. **(5 marks)**

An increase in wind_speed by one unit is associated with an estimated multiplicative change of 0.1049 in the odds of a fire occurring, adjusting for other predictors.

3. (25 marks)

a. **(5 marks)**

The optimal model chosen by forward selection is FIRE ~ DC + WIND_SPEED + X. (principle of parsimony would however pick FIRE ~ DC as the gain in AIC between each stage of the forward selection is very little)

The optimal model chosen by backward selection is FIRE ~ WIND_SPEED + X + SEASON + FFMC + TEMPERATURE.

The optimal model chosen by backward selection is much more complex. The two techniques can often yield different models due to the differences in the algorithms and their stopping criteria.

b. **(5 marks)**

When using test error rate as the selection criteria, the goal is to find a model that minimises the test error rate. In this instance, the model with the lowest error rate is FIRE ~ SEASON + DC + TEMPERATURE, however since all the models listed are within one standard error with one another then the model chosen by test error selection is FIRE ~ SEASON + TEMPERATURE as that's the simplest model with an error rate still within one standard deviation of the best model (FIRE ~ DC + WIND_SPEED has the same complexity and also within one standard deviation of the best model but has a higher test rate than FIRE ~ SEASON + TEMPERATURE).

When using AUC, the best model is one that maximises AUC, which is the model that is FIRE ~ SEASON + TEMPERATURE, the same model selected via test error.

c. **(5 marks)**

Repeated 5-fold/10-fold cross-validation is generally the preferred model selection method because it's useful when your dataset is small where a basic test/train split approach wouldn't be advised or when the ordering of the instances can introduce biases into the model (ie. if you have a lot of one class and not many of the other and you test/train split leaving you with a test or train set that has all of 1 class). It's also useful to measure the predictive performance over a range of models.

The reason why it's not always used is that it can be extremely computationally expensive especially if you have a lot of predictors

d. **(5 marks)**

Standardising is used to scale the data so that comparisons between predictors on different scales is compared correctly. Without standardising a range of one predictor could be in the 10000s whilst another could be in the 100s, standardising brings these predictors in line with one another so that one predictor does not significantly skew an instance when calculating and comparing distances.

e. **(5 marks)**

The squared mahalanobis distance Q-Q plots have major deviations from the line suggesting that there's strong evidence that there's departure from mahalanobis normality which suggest QDA is more appropriate than LDA.

The Box's M test yields a significantly low p-value of 2.144e-11 which is much lower than any reasonable confidence threshold which gives very strong evidence of a violation of the equal variances and covariances assumption for linear discriminant analysis and QDA would be much preferred.

However the normal Q-Q plots suggests that the data comes from a distribution that isn't normally distributed and thus the results of the QDA will be unreliable.

Based on the above a knn would be more appropriate as knn is a non-parametric classification model so it doesn't need to make any assumptions about the underlying data.

Total: 110 marks