MATH 3190 Homework 6

Regularization, Cross-validation, Dimension reduction

Due 4/11/2022

In this homework you will practice using cross-validation to fit data using LASSO and K-nearest neighbor models. Please upload to your GitHub an R Markdown document answering the following:

\begin{enumerate}
(20 points) A researcher wants to determine how employee salaries at a certain company are related to the length of employment, previous experience, and education. The researcher selects eight employees from the company and obtains the data shown below (the dataset is available as a tibble in the .Rmd).

```
\begin{enumerate}
\item Fit a standard least squares regression model to these data and interpret the results. After looking at the statistical significance of the $\beta$s, which covariates would you include in a final model?
```

```
employment_mod <- lm(Salary ~., data = salary)
summary(employment_mod)
```

```
##
## Call:
## lm(formula = Salary ~ ., data = salary)
##
## Residuals:
##       1       2       3       4       5       6       7       8
## -824.76  156.82 -153.52  158.90  -56.65  364.09  804.95 -449.82
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 49764.45    1981.35  25.116 1.49e-05 ***
## Employment    364.41      48.32   7.542  0.00166 **
## Experience    227.62     123.84   1.838  0.13991
## Education     266.94     147.36   1.812  0.14430
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 659.5 on 4 degrees of freedom
## Multiple R-squared:  0.9438, Adjusted R-squared:  0.9017
## F-statistic:  22.4 on 3 and 4 DF,  p-value: 0.005804
```

salary = 49745 + 364.41 * Employment + 227.62 * Experience + 226.94 * Education + error

```
 cor(salary)
```

```
##               Salary Employment Experience  Education
## Salary     1.0000000  0.8238657  0.1892612  0.3746145
## Employment 0.8238657  1.0000000 -0.3102595 -0.1082163
## Experience 0.1892612 -0.3102595  1.0000000  0.6452144
## Education  0.3746145 -0.1082163  0.6452144  1.0000000
```

Employment for sure should stay, hard to say about experience and education. THey do seem correlated with each other. They're pretty close in terms of p value.
Use to fit a LASSO model to these covariates. Try λ=1000, 800, 500, and 1. How do the results compare to each other and the least squares model?

```
y <- salary$Salary
x <- data.matrix(salary[, c('Employment', 'Experience', 'Education')])
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.0.5
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-3
```

```
lasso1000 <- glmnet(x,y, alpha = 1, lambda = 1000)
lasso800 <- glmnet(x,y, alpha = 1, lambda = 800)
lasso500 <- glmnet(x,y, alpha = 1, lambda = 500)
lasso1 <- glmnet(x,y, alpha = 1, lambda = 1)
lasso_salary <- glmnet(x,y, alpha = 1)

lasso1000
```

```
##
## Call:  glmnet(x = x, y = y, alpha = 1, lambda = 1000)
##
##   Df  %Dev Lambda
## 1  1 42.05   1000
```

```
lasso800
```

```
##
## Call:  glmnet(x = x, y = y, alpha = 1, lambda = 800)
##
##   Df  %Dev Lambda
## 1  2 52.56    800
```

```
lasso500
```

```
##
## Call:  glmnet(x = x, y = y, alpha = 1, lambda = 500)
##
##   Df  %Dev Lambda
## 1  2 75.16    500
```

```
lasso1
```

```
##
## Call:  glmnet(x = x, y = y, alpha = 1, lambda = 1)
##
##   Df  %Dev Lambda
## 1  3 94.38      1
```

```
lasso_salary
```

```
##
## Call:  glmnet(x = x, y = y, alpha = 1)
##
##    Df  %Dev  Lambda
## 1   0  0.00 1621.00
## 2   1 11.52 1477.00
## 3   1 21.09 1346.00
## 4   1 29.03 1226.00
## 5   1 35.63 1117.00
## 6   1 41.10 1018.00
## 7   1 45.65  927.60
## 8   1 49.42  845.20
## 9   2 55.28  770.10
## 10  2 61.11  701.70
## 11  2 65.96  639.40
## 12  2 69.98  582.60
## 13  2 73.32  530.80
## 14  3 76.34  483.70
## 15  3 79.41  440.70
## 16  3 81.95  401.50
## 17  3 84.06  365.90
## 18  3 85.81  333.40
## 19  3 87.27  303.80
## 20  3 88.47  276.80
## 21  3 89.48  252.20
## 22  3 90.31  229.80
## 23  3 91.00  209.40
## 24  3 91.58  190.80
## 25  3 92.05  173.80
## 26  3 92.45  158.40
## 27  3 92.78  144.30
## 28  3 93.05  131.50
## 29  3 93.28  119.80
## 30  3 93.46  109.20
## 31  3 93.62   99.47
## 32  3 93.75   90.63
## 33  3 93.86   82.58
## 34  3 93.95   75.24
## 35  3 94.02   68.56
## 36  3 94.08   62.47
## 37  3 94.13   56.92
## 38  3 94.18   51.86
## 39  3 94.21   47.25
## 40  3 94.24   43.06
## 41  3 94.26   39.23
## 42  3 94.28   35.75
## 43  3 94.30   32.57
## 44  3 94.31   29.68
## 45  3 94.33   27.04
## 46  3 94.34   24.64
## 47  3 94.34   22.45
## 48  3 94.35   20.46
## 49  3 94.36   18.64
## 50  3 94.36   16.98
## 51  3 94.36   15.47
## 52  3 94.37   14.10
## 53  3 94.37   12.85
## 54  3 94.37   11.71
## 55  3 94.37   10.67
## 56  3 94.38    9.72
## 57  3 94.38    8.86
## 58  3 94.38    8.07
## 59  3 94.38    7.35
```

```
\item Which LASSO model (i.e. $\lambda$) would you select? (note you are not just restricted to $\lambda$ values of 1000, 800, 500, and 1). Justify your answer.
```

I like a lambda of 500. Looking at the many values of lambda with the lasso_salary model, the best one inclued in that output has a lambda of 530.8, that explains 73.72% of the variation, while the lambda of 500 uses the same 2 and explains 75.16% of it. I would go with the 500 lambda model since it seems like the best I can get without introducing multicollinearity from the experience and education models.

```
\item Use \texttt{glmnet} to fit a Ridge regression model to these data. Try $\lambda$=1000, 800, 500, and 1. How do these results differ from the least squares and LASSO models?
\end{enumerate}
```

```
ridge1000 <- glmnet(x,y, alpha = 0, lambda = 1000)
ridge800 <- glmnet(x,y, alpha = 0, lambda = 800)
ridge500 <- glmnet(x,y, alpha = 0, lambda = 500)
ridge1 <- glmnet(x,y, alpha = 0, lambda = 1)


ridge1000
```

```
##
## Call:  glmnet(x = x, y = y, alpha = 0, lambda = 1000)
##
##   Df %Dev Lambda
## 1  3 81.9   1000
```

```
ridge800
```

```
##
## Call:  glmnet(x = x, y = y, alpha = 0, lambda = 800)
##
##   Df  %Dev Lambda
## 1  3 85.05    800
```

```
ridge500
```

```
##
## Call:  glmnet(x = x, y = y, alpha = 0, lambda = 500)
##
##   Df  %Dev Lambda
## 1  3 89.65    500
```

```
ridge1
```

```
##
## Call:  glmnet(x = x, y = y, alpha = 0, lambda = 1)
##
##   Df  %Dev Lambda
## 1  3 94.38      1
```

```
coef(ridge1000)
```

```
## 4 x 1 sparse Matrix of class "dgCMatrix"
##                     s0
## (Intercept) 52287.3870
## Employment    228.1970
## Experience    114.1154
## Education     213.4515
```

```
coef(ridge800)
```

```
## 4 x 1 sparse Matrix of class "dgCMatrix"
##                     s0
## (Intercept) 51888.0446
## Employment    246.3775
## Experience    125.7664
## Education     225.6544
```

```
coef(ridge500)
```

```
## 4 x 1 sparse Matrix of class "dgCMatrix"
##                     s0
## (Intercept) 51189.1133
## Employment    279.9839
## Experience    149.4110
## Education     245.0589
```

```
coef(ridge1)
```

```
## 4 x 1 sparse Matrix of class "dgCMatrix"
##                     s0
## (Intercept) 49767.5944
## Employment    364.2004
## Experience    227.4089
## Education     266.9118
```

Since the Ridge doesn't take any coeficents to 0, I think I would just use the lambda of 1 here that expalins the most variation. While the values of the coeficents are reduced, I don't think there's enough redcution to take out the multicollinearity problem.

(20 points) The dataset provides nutritional information on nearly 80 common breakfast cereals. The 'rating' column provides an overall rating for each cereal (possibly from Consumer Reports?). Use a LASSO regression model to identify the best predictors of cereal rating. Evaluate the model for λ values of 8, 5, 3, and 1 (among others). Which λ would you choose and why? Which covariates best explain the rating?

```
cereal <- read.csv("cereal.csv")

cereal8 <- glmnet(data.matrix(cereal[,2:15 ]), cereal$rating, alpha = 1, lambda = 8)

cereal5 <- glmnet(data.matrix(cereal[,2:15 ]), cereal$rating, alpha = 1, lambda = 5)

cereal3 <- glmnet(data.matrix(cereal[,2:15 ]), cereal$rating, alpha = 1, lambda = 3)

cereal1 <- glmnet(data.matrix(cereal[,2:15 ]), cereal$rating, alpha = 1, lambda = 1)

cereal8
```

```
##
## Call:  glmnet(x = data.matrix(cereal[, 2:15]), y = cereal$rating, alpha = 1,      lambda = 8)
##
##   Df %Dev Lambda
## 1  2 25.7      8
```

```
cereal5
```

```
##
## Call:  glmnet(x = data.matrix(cereal[, 2:15]), y = cereal$rating, alpha = 1,      lambda = 5)
##
##   Df  %Dev Lambda
## 1  3 61.08      5
```

```
cereal3
```

```
##
## Call:  glmnet(x = data.matrix(cereal[, 2:15]), y = cereal$rating, alpha = 1,      lambda = 3)
##
##   Df %Dev Lambda
## 1  5 80.6      3
```

```
cereal1
```

```
##
## Call:  glmnet(x = data.matrix(cereal[, 2:15]), y = cereal$rating, alpha = 1,      lambda = 1)
##
##   Df  %Dev Lambda
## 1  7 95.12      1
```

```
coef(cereal1)
```

```
## 15 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept) 60.1312767244
## mfr          .
## type         .
## calories    -0.0462127277
## protein      1.0136726660
## fat         -2.7207952034
## sodium      -0.0385419678
## fiber        2.1861429137
## carbo        .
## sugars      -1.5716905511
## potass       .
## vitamins    -0.0007519146
## shelf        .
## weight       .
## cups         .
```

I like the lambda = 1 model here. It still takes some variables to 0 while explaining 95.12% of the variation. I might not include vitamins since the coefficient is so small in comparison to the magnitude of the ratings variable, but I would keep calories, protein, fat, sodium, fiber, and sugars.

(20 points) An automobile consulting company wants to understand the factors on which the pricing of cars depends. Use an Elastic Net model and the dataset to determine which variables are significant in predicting the price of a car. Use cross-validation to find an optimal value for λ. Interpret your final model.

```
car_price <- read.csv("car_price_prediction.csv")

car_cv <- cv.glmnet (data.matrix(car_price[,c(1,2,4:8)]), car_price$selling_price)

car_best_lambda_model <- glmnet(data.matrix(car_price[,c(1,2,4:8)]), car_price$selling_price, lambda = car_cv$lam
bda.min)

coef(car_best_lambda_model)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept)  -7.107859e+07
## name         -4.057454e+01
## year          3.662590e+04
## km_driven    -9.409250e-01
## fuel         -9.257513e+04
## seller_type  -1.817989e+04
## transmission -8.814019e+05
## owner        -1.657610e+04
```

```
car_best_lambda_model
```

```
##
## Call:  glmnet(x = data.matrix(car_price[, c(1, 2, 4:8)]), y = car_price$selling_price,      lambda = car_cv$la
mbda.min)
##
##   Df  %Dev Lambda
## 1  7 44.98   1155
```

```
car_lasso <- glmnet(data.matrix(car_price[,c(1,2,4:8)]), car_price$selling_price, lambda = car_cv$lambda.min, alp
ha = 1)
car_lasso
```

```
##
## Call:  glmnet(x = data.matrix(car_price[, c(1, 2, 4:8)]), y = car_price$selling_price,      alpha = 1, lambda
= car_cv$lambda.min)
##
##   Df  %Dev Lambda
## 1  7 44.98   1155
```

```
coef(car_best_lambda_model)
```

```
## 8 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept)  -7.107859e+07
## name         -4.057454e+01
## year          3.662590e+04
## km_driven    -9.409250e-01
## fuel         -9.257513e+04
## seller_type  -1.817989e+04
## transmission -8.814019e+05
## owner        -1.657610e+04
```

```
summary(lm(selling_price~. - name, data = car_price))
```

```
##
## Call:
## lm(formula = selling_price ~ . - name, data = car_price)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1185295  -166741   -23884   114047  7547813
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 -6.971e+07  3.872e+06 -18.006  < 2e-16 ***
## year                         3.526e+04  1.918e+03  18.379  < 2e-16 ***
## km_driven                   -9.591e-01  1.683e-01  -5.699 1.28e-08 ***
## fuelDiesel                   2.863e+05  6.818e+04   4.200 2.72e-05 ***
## fuelElectric                -6.059e+05  4.324e+05  -1.401 0.161172
## fuelLPG                      4.700e+04  1.117e+05   0.421 0.673889
## fuelPetrol                  -4.245e+03  6.823e+04  -0.062 0.950391
## seller_typeIndividual       -6.638e+04  1.648e+04  -4.029 5.70e-05 ***
## seller_typeTrustmark Dealer  1.675e+05  4.446e+04   3.768 0.000167 ***
## transmissionManual          -8.703e+05  2.202e+04 -39.533  < 2e-16 ***
## ownerFourth & Above Owner   -1.454e+03  4.986e+04  -0.029 0.976729
## ownerSecond Owner           -4.093e+04  1.668e+04  -2.454 0.014157 *
## ownerTest Drive Car          1.687e+05  1.048e+05   1.609 0.107656
## ownerThird Owner            -3.993e+04  2.778e+04  -1.437 0.150751
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 426100 on 4326 degrees of freedom
## Multiple R-squared:  0.4593, Adjusted R-squared:  0.4576
## F-statistic: 282.6 on 13 and 4326 DF,  p-value: < 2.2e-16
```

All the variables in this data appear to be significant. From the car_best_lambda model since year has a positive coefficient each year a vehicle gets older, the expected price goes down by $36626. For each additional km driven the expected price goes down by -$0.94. Based on the model summary, the fuel being LPG is what decreases the expected price by $92,575 compared to CNG. The seller being an individual reduces the expected price by $18180 compared to being sold by a dealer, the transmission being manual reduces the expected price by $881,402, and being the 4th owner or above leads to an expected decrease in price of $16,576

These coefficients all seem large, and looking through the dataset, the prices are all way too high, so it is likely the they are not in US dollars. For the categorical variables I looked at the basic linear model to know what the comparison category is, and what the omitted category is.

\end{enumerate}