Module 4 - Supervised

Terry Zhou

3/28/2022

Warm Up

1. Linear regression can be used in machine learning for predictive modeling. It is used to minimize the error of a model and improve predictions.
2. Lasso and Ridge Regrssion can be used to reduce model complexity and prevent over0fitting. Ridge regression decreases the coefficients and complexity of the model. Lasso regression reduces over-fitting and assists in feature selection.
3. One hot coding is when categorical variables are converted into a form that machine learning algorithms can use to improve their prediction.
4. R squared is the correlation coefficient and RMSE is the residual mean squared error.
5. SVMs are support vector machines. They are used for classification, regression, and outliers detection. KNN is k-nearest neighbors algorithm and is a nonparametric supervised learning method. It is used for classification and regression. LDA is linear discriminant analysis and is used for classification, dimension reduction, and data visualization. Logistic regression is a statistical model that uses a logistic function to model the dependent variable.
6. K-fold cross validation is used to estimate the skill of the model on new data. The k parameter refers to the number of groups that a given dataset will be split into.
7. input layer, hidden layer, output layer

Classification

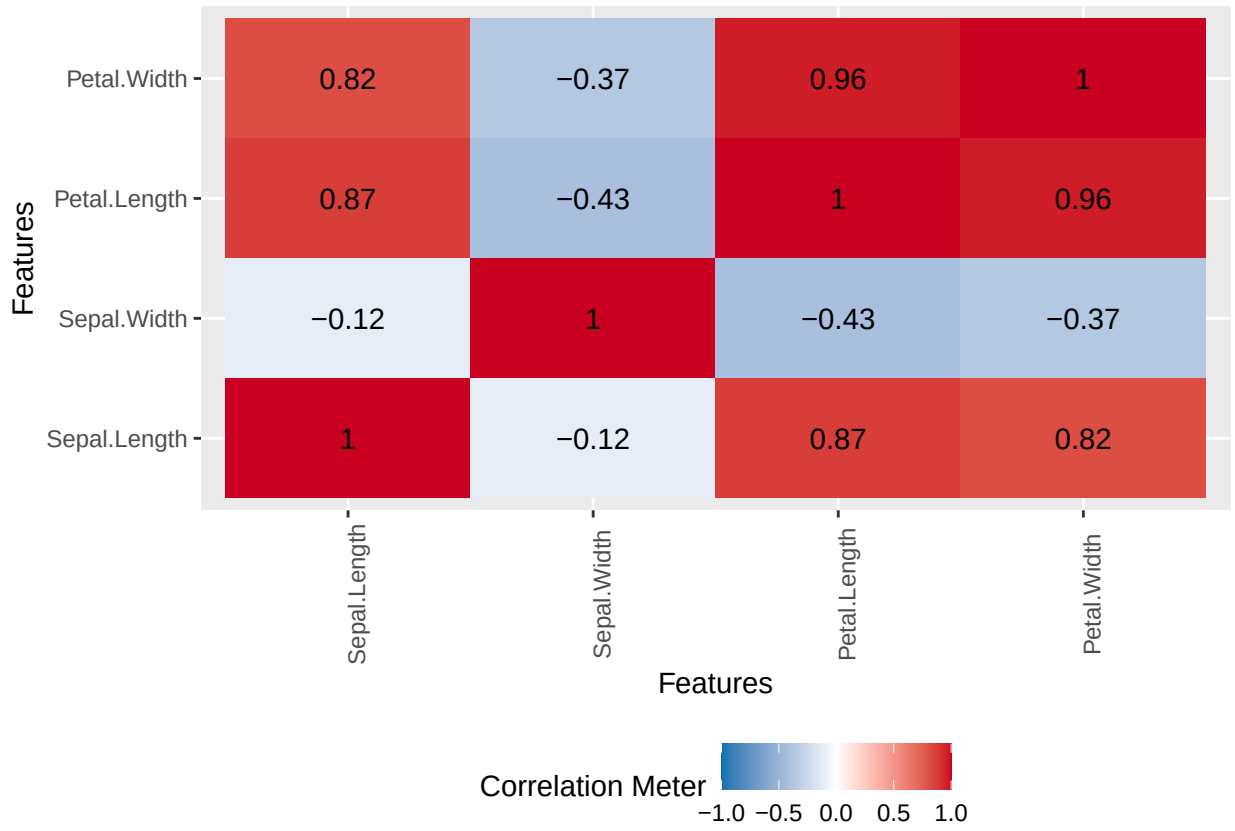
1.

```
data(iris)
summary(iris)
```

```
##   Sepal.Length    Sepal.Width     Petal.Length    Petal.Width
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
##        Species
##  setosa    :50
```

```
##  versicolor:50
##  virginica :50
##
##
##
```

```
library(DataExplorer)
plot_correlation(iris[, c(1:4)])
```



2.

```
n <- floor(0.70 * nrow(iris))
set.seed(123)
ind <- sample(seq_len(nrow(iris)), size = n)
train <- iris[ind, ]
test <- iris[-ind, ]

means <- apply( X = train, MARGIN = 2, FUN = mean )
```

```
## Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(newX[, i], ...): argument is not numeric or logical:
## returning NA
```

```
std <- apply( X = train, MARGIN = 2, FUN = sd )
```

```
## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
scale1 <- test %>%
  sweep( MARGIN = 2, STATS = means, FUN = "-" ) %>%
  sweep( MARGIN = 2, STATS = std, FUN = "/" )
```

```
## Warning in Ops.factor(left, right): '-' not meaningful for factors
```

3.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.1.4     v forcats 0.5.1
## v readr   2.1.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(reticulate)
```

Regression

1.

```
insurance <- read.csv("insurance.csv")
nrow(insurance)
```

```
## [1] 1338
```

There are 1338 observations.

2.

```
str(insurance)
```

```
## 'data.frame':    1338 obs. of  7 variables:
##  $ age     : int  19 18 28 33 32 31 46 37 37 60 ...
##  $ sex     : chr  "female" "male" "male" "male" ...
##  $ bmi     : num  27.9 33.8 33 22.7 28.9 ...
##  $ children: int  0 1 3 0 0 0 1 3 2 0 ...
##  $ smoker  : chr  "yes" "no" "no" "no" ...
##  $ region  : chr  "southwest" "southeast" "southeast" "northwest" ...
##  $ charges : num  16885 1726 4449 21984 3867 ...
```

There are 7 variables. The age, charges, BMI, and children variables are numerical and the sex, smoker, and region variables are categorical.

3.

```
apply(is.na(insurance), 2, which)
```

```
## $age
## integer(0)
##
## $sex
## integer(0)
##
## $bmi
## [1]  7 19
##
## $children
## integer(0)
##
## $smoker
```

```
## integer(0)
##
## $region
## integer(0)
##
## $charges
## [1] 24 28 51 61
```

```
insurance1 <- insurance[-c(7, 19, 24,28, 51, 61), ]
```

There are missing values.

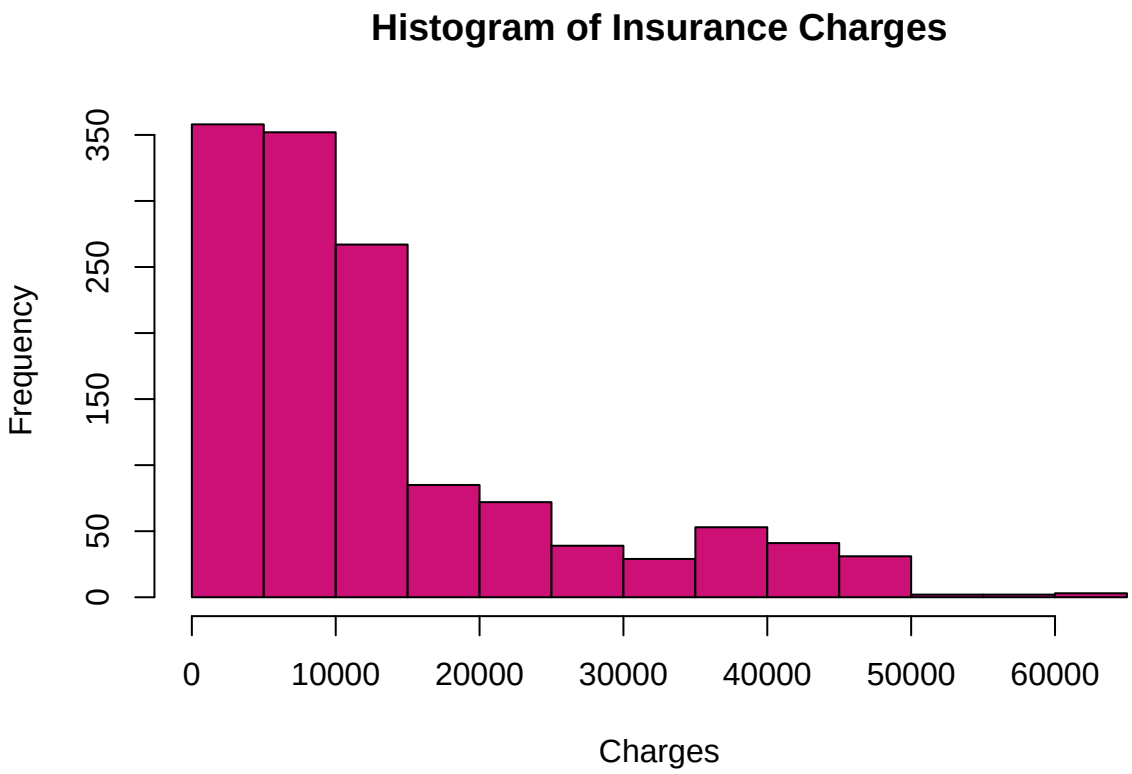
4.

```
summary(insurance1[, c(1, 2, 4, 7)])
```

```
##       age            sex                children        charges
##  Min.   :18.00   Length:1332        Min.   :0.000   Min.   : 1122
##  1st Qu.:26.75   Class :character   1st Qu.:0.000   1st Qu.: 4734
##  Median :39.00   Mode  :character   Median :1.000   Median : 9382
##  Mean   :39.19                      Mean   :1.095   Mean   :13270
##  3rd Qu.:51.00                      3rd Qu.:2.000   3rd Qu.:16687
##  Max.   :64.00                      Max.   :5.000   Max.   :63770
```

5.

```
hist(insurance$charges, main = "Histogram of Insurance Charges",
     xlab = "Charges", col = "deeppink3")
```



The distribution of charges is skew right and not normally distributed. I would log transform the data so that it is more normal.

6.

```
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

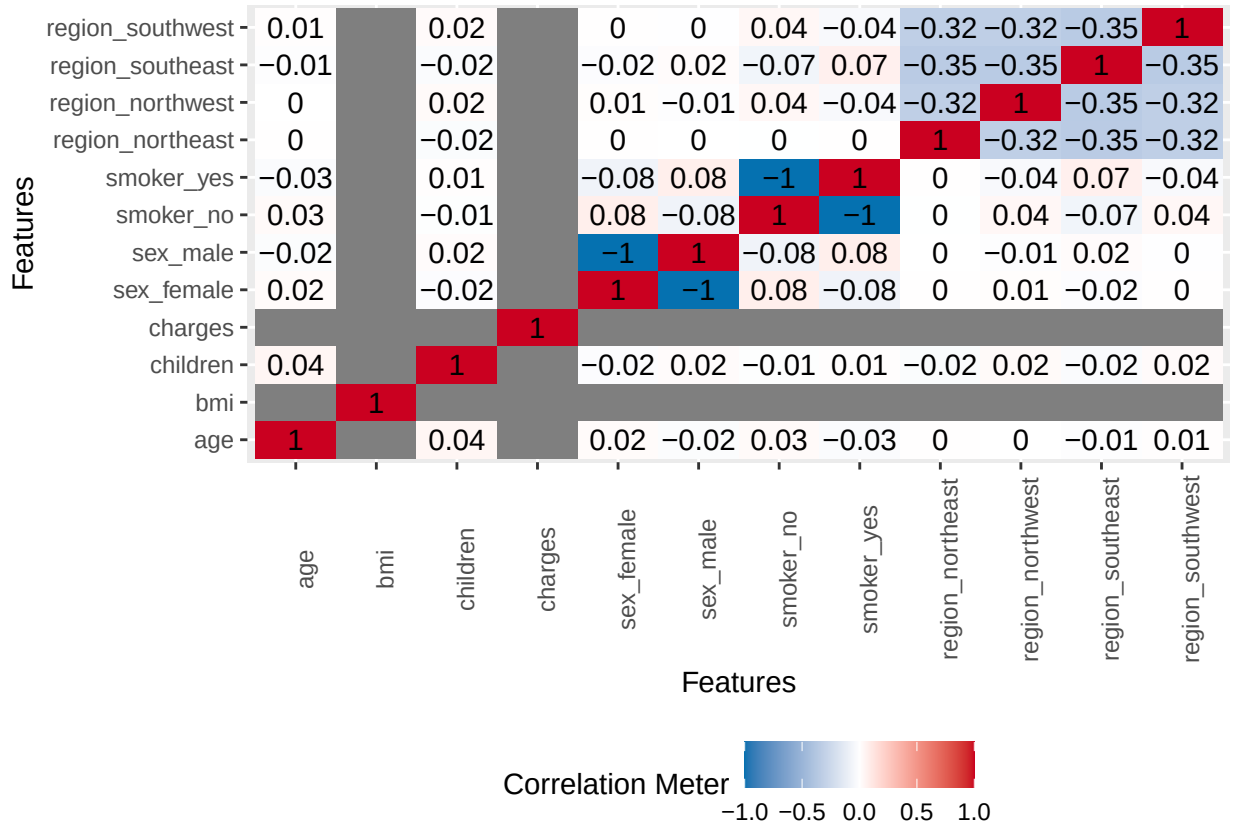
```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
dummy <- dummyVars(" ~ .", data=insurance1)
insurance2 <- data.frame(predict(dummy, newdata = insurance1))
```

7.

```
plot_correlation(insurance)
```

```
## Warning: Removed 42 rows containing missing values (geom_text).
```



None of the variables show a strong correlation with charges.

8.

```
n.1 <- floor(0.70 * nrow(insurance2))
set.seed(132)
ind1 <- sample(seq_len(nrow(insurance2)), size = n.1)
train.i <- insurance2[ind1, ]
test.i <- insurance2[-ind1, ]

means1 <- apply( X = train.i, MARGIN = 2, FUN = mean )
std1 <- apply( X = train.i, MARGIN = 2, FUN = sd )
scale.i <- test.i %>%
  sweep( MARGIN = 2, STATS = means1, FUN = "-" ) %>%
  sweep( MARGIN = 2, STATS = std1, FUN = "/" )
```

9.

```
# linear
lin <- lm(charges ~ age + sexfemale + sexmale + bmi + children + smokerno +
            smokeryes + regionnortheast + regionnorthwest + regionsoutheast +
            regionsouthwest, data = train.i)
summary(lin)
```

```
##
## Call:
## lm(formula = charges ~ age + sexfemale + sexmale + bmi + children +
##     smokerno + smokeryes + regionnortheast + regionnorthwest +
##     regionsoutheast + regionsouthwest, data = train.i)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10995.7  -2853.3   -879.4   1463.3  29829.6
##
## Coefficients: (3 not defined because of singularities)
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      10097.66    1270.62   7.947 5.55e-15 ***
## age                255.36      14.18  18.007  < 2e-16 ***
## sexfemale          182.17     394.22   0.462   0.6441
## sexmale                NA         NA      NA       NA
## bmi                355.06      34.33  10.342  < 2e-16 ***
## children           438.14     162.96   2.689   0.0073 **
## smokerno        -23889.99     484.70 -49.288  < 2e-16 ***
## smokeryes              NA         NA      NA       NA
## regionnortheast   1228.61     571.94   2.148   0.0320 *
## regionnorthwest    961.35     556.80   1.727   0.0846 .
## regionsoutheast    394.77     556.84   0.709   0.4785
## regionsouthwest        NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5990 on 923 degrees of freedom
## Multiple R-squared:  0.7636, Adjusted R-squared:  0.7616
## F-statistic: 372.7 on 8 and 923 DF,  p-value: < 2.2e-16
```

```
# ridge
y <- train.i$charges
x <- data.matrix(train.i[, -c(5)])
library(glmnet)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

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

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

```
ridge <- glmnet(x, y, alpha = 0)
summary(ridge)
```

```
##           Length Class     Mode
## a0         100   -none-    numeric
## beta      1100   dgCMatrix S4
## df         100   -none-    numeric
## dim          2   -none-    numeric
## lambda     100   -none-    numeric
## dev.ratio  100   -none-    numeric
## nulldev      1   -none-    numeric
## npasses      1   -none-    numeric
## jerr         1   -none-    numeric
## offset       1   -none-    logical
## call         4   -none-    call
## nobs         1   -none-    numeric
```

```
# lasso
cv.lasso <- cv.glmnet(x, y, alpha = 1)
best.lambda <- cv.lasso$lambda.min
best.lambda
```

```
## [1] 357.3967
```

```
lasso <- glmnet(x, y, alpha = 1, lambda = best.lambda)
```

10.

```
test.lin = function(model, df, predictions, target){
    resids = df[,target] - predictions
    resids2 = resids**2
    n = length(predictions)
    r2 = as.character(round(summary(model)$r.squared, 2))
    print(r2)
}
predict.lin <- predict(lin, newdata = test.i)
```

```
## Warning in predict.lm(lin, newdata = test.i): prediction from a rank-deficient
## fit may be misleading
```

```
test.lin(lin, test.i, predict.lin, target = 'charges')
```

```
## [1] "0.76"
```

```
test.ridge <- function(true, predicted, df) {
  SSE <- sum((predicted - true)^2)
  SST <- sum((true - mean(true))^2)
  r2 <- 1 - SSE / SST
  print(r2)
}
x.test <- as.matrix(test.i[, -c(5)])
lambdas <- 10^seq(2, -3, by = -.1)
cv_ridge <- cv.glmnet(x, y, alpha = 0, lambda = lambdas)
optimal_lambda <- cv_ridge$lambda.min
optimal_lambda
```

```
## [1] 0.001
```

```
predict.ridge <- predict(ridge, s = optimal_lambda, newx = x.test)
test.ridge(test.i$charges, predict.ridge, test.i)
```

```
## [1] 0.9799037
```

```
predict.lasso <- predict(lasso, s = best.lambda, newx = x)
sst <- sum((y - mean(y))^2)
sse <- sum((predict.lasso - y)^2)
r2 <- 1 - sse/sst
r2
```

```
## [1] 0.9991502
```

The lasso regression R squared value is 0.9991502, the ridge regression R squared is 0.9799037, and the linear regression R squared value is 0.76.