Support Vector Machines and Neural Networks

Purpose of this Project:

This project involves building SVM and neural network regression models to answer a number of questions. We will use the Credit dataset that is part of the ISLR package.

Loading required libraries:

```
library(ISLR)
library(dplyr)
library(glmnet)
library(caret)
```

Selecting required features:

```
library(dplyr)
credit_Filtered = Credit[, c("Income", "Limit", "Rating", "Cards", "Age", "Balance")]
summary(credit_Filtered)
```

```
##      Income           Limit           Rating          Cards
##  Min.   : 10.35   Min.   :  855   Min.   : 93.0   Min.   :1.000
##  1st Qu.: 21.01   1st Qu.: 3088   1st Qu.:247.2   1st Qu.:2.000
##  Median : 33.12   Median : 4622   Median :344.0   Median :3.000
##  Mean   : 45.22   Mean   : 4736   Mean   :354.9   Mean   :2.958
##  3rd Qu.: 57.47   3rd Qu.: 5873   3rd Qu.:437.2   3rd Qu.:4.000
##  Max.   :186.63   Max.   :13913   Max.   :982.0   Max.   :9.000
##       Age           Balance
##  Min.   :23.00   Min.   :   0.00
##  1st Qu.:41.75   1st Qu.:  68.75
##  Median :56.00   Median : 459.50
##  Mean   :55.67   Mean   : 520.01
##  3rd Qu.:70.00   3rd Qu.: 863.00
##  Max.   :98.00   Max.   :1999.00
```

Building a linear SVM regression model to predict Sales based on all other attributes ("Income", "Limit", "Rating", "Cards", "Age", "Balance")):

```
set.seed(1203)
SVM_Model<- train(Income~.,data=credit_Filtered,method="svmLinear",preProcess=c("center","scale"),tuneL
SVM_Model
```

```
## Support Vector Machines with Linear Kernel
##
## 400 samples
##   5 predictor
##
## Pre-processing: centered (5), scaled (5)
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 400, 400, 400, 400, 400, 400, ...
## Resampling results:
##
```

```
##   RMSE      Rsquared   MAE
##   17.01727  0.8008214  11.59274
##
## Tuning parameter 'C' was held constant at a value of 1
```

R square is 80.08%

Customizing the search grid by checking the model's performance for C parameter of 0.1,.5,1 and 10 using 2 repeats of 5-fold cross validation.

```
set.seed(1203)

grid = expand.grid(C= c(0.1,0.5,1,10))

trctrl <- trainControl(method = "repeatedcv", number = 5, repeats = 2)

SVM_Model2 <- train(Income ~., data = credit_Filtered, method = "svmLinear",
trControl=trctrl,
preProcess = c("center", "scale"), tuneGrid = grid,
tuneLength = 10)

SVM_Model2
```

```
## Support Vector Machines with Linear Kernel
##
## 400 samples
##   5 predictor
##
## Pre-processing: centered (5), scaled (5)
## Resampling: Cross-Validated (5 fold, repeated 2 times)
## Summary of sample sizes: 320, 320, 320, 320, 320, 320, ...
## Resampling results across tuning parameters:
##
##   C     RMSE      Rsquared   MAE
##    0.1  15.96848  0.8147851  12.22546
##    0.5  17.12632  0.8112791  11.75740
##    1.0  17.39896  0.8104476  11.69149
##   10.0  17.65612  0.8087640  11.65906
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was C = 0.1.
```

Training a neural network model to predict Sales based on all other attributes ("Price", "Advertising", "Population", "Age", "Income" and "Education").

```
Normalization <- preProcess(credit_Filtered[,1:4],method = c("center","scale"))
Norm_data<-predict(Normalization,credit_Filtered)

NNET_Model<- train(Income~.,data=Norm_data,method="nnet",linout=TRUE, trace = FALSE)
NNET_Model
```

```
## Neural Network
##
## 400 samples
##   5 predictor
##
## No pre-processing
```

```
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 400, 400, 400, 400, 400, 400, ...
## Resampling results across tuning parameters:
##
##   size  decay  RMSE       Rsquared   MAE
##   1     0e+00  0.7774903  0.3789332  0.5712278
##   1     1e-04  0.7979874  0.3860290  0.5933617
##   1     1e-01  0.3946719  0.7959692  0.2844267
##   3     0e+00  0.5621655  0.6211083  0.4086901
##   3     1e-04  0.5064937  0.6636561  0.3650899
##   3     1e-01  0.3412299  0.8666945  0.2431175
##   5     0e+00  0.4921719  0.6868286  0.3542473
##   5     1e-04  0.4051493  0.7933676  0.2873787
##   5     1e-01  0.3129288  0.9020300  0.2210356
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were size = 5 and decay = 0.1.
```

R-square value with best hyperparameters(size=1) is 37.89.

Consider the following input: (Limit = 9, Age = 76, Income = 110, Rating = 100, Cards = 1, Balance = 100). Estimating Sales for this record using the above neuralnet model?

```
Input <- data.frame(Limit = 9, Age = 76, Income = 110, Rating = 100, Cards = 1,  Balance = 100)

Prediction<- predict(NNET_Model,Input )
Prediction
```

```
##        1
## 4.669901
```

Estimated Income: 4.669901