Learning outcomes:

Training Neural Networks on multiple data sets for

- Classification problem (Demo)
- Regression problem
- Digit recognition problem

1. A marketing department of a bank runs various marketing campaigns for cross-selling products, improving customer retention and better customer services. Let us take a marketing data sample in which the bank is interested in selling a term deposit product to its existing customers.
   Contacting all customers is costly and does not create good customer experience. So, the bank wants to build a predictive model which will identify customers who are more likely to respond to the marketing campaign.

2. The dataset "housing.csv" has information on the housing values in suburbs of Boston. Predict the median value of the owner occupied homes "OwnOcc" using neural net algorithm. The attributes description is given below:

   - Crimerate: per capita crime rate by town
   - ResiLandZone: proportion of residential land zoned for lots over 25,000 sq.ft.
   - INDUS: proportion of nonretail business acres per town
   - CHAS: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
   - NOX: nitric oxides concentration (parts per 10 million)
   - Rooms: average number of rooms per dwelling
   - AGE: proportion of owner occupied units built prior to 1940
   - Distance: weighted distances to five Boston employment centers
   - RAD: index of accessibility to radial highways
   - TAX: full value property tax rate per $10,000
   - PTRATIO: pupil teacher ratio by town
   - Blacks: 1000(Bk 0.63)^2 where Bk is the proportion of blacks by town
   - LSTAT: % lower status of the population
     OwnOcc: Median value of owner occupied homes in $1000's

Steps

a) Read "housing.csv" file into R
b) install.packages("drat", repos="https://cran.rstudio.com")
   #drat:::addRepo("dmlc")
   #install.packages("mxnet")
c) Study structure, summary etc.
d) Apply type conversion to attributes if necessary.
e) If attributes are categorical, convert them into dummies.

f) Separate target attribute and independent attributes
g) Standardize the independent attributes using “range” method
h) Combine the target attribute and the standardized independent attributes to create the final dataframe.
i) Split dataset into train and test as follows:

```
library (caret)
set.seed (1234)
intrain = createDataPartition (y = dataframe_name$OwnOcc, p=0.7, list = F)
train.x = data.matrix (dataframe_name [intrain, -15])
train.y = dataframe_name [intrain, 15]
test.x = data.matrix (dataframe_name [-intrain, -15])
test.y = dataframe_name [-intrain, 15]
```

j) require(mxnet)
k) mx.set.seed(0)
l) model <- mx.mlp (train.x, train.y, hidden_node=10, out_node=1,
activation="tanh",out_activation="rmse", num.round=20,
array.batch.size=100, learning.rate=0.07, momentum=0.9,
eval.metric=mx.metric.rmse)
m) Predict with test data
n) Compute error metrics

3. Digit Recognition problem. These dataset have images corresponding to 10 digits(0-9) Each sample in the dataset is a greyscale image of 28x28 resolution. Consider each pixel as a variable and train the neural network to classify samples.

```
# Load train and test datasets
train <- read.csv("train_sample.csv")
test <- read.csv("test_sample.csv")

# Convert to matrix
train_mat <- data.matrix(train)

# dataset into train and test
    train<-data.matrix(train)
    test<-data.matrix(test)

    train.x<-train[,-1]
    train.y<-train[,1]
    train.x<-(train.x/255)
```

```
test.x<- test [,-1]
test.y<- test [,1]
test.x<-(test.x/255)

#Build model
model1<-mx.mlp(train.x, train.y, hidden_node =c(128,64), out_node=10,
dropout = NULL, activation = "relu", out_activation = "softmax",
      num.round=10, array.batch.size=100, learning.rate=0.07,
      momentum=0.9, eval.metric=mx.metric.accuracy)

# Predict on test
preds <- predict(model1, test.x)
dim(preds)
pred.label <- max.col(t(preds)) - 1
table(pred.label)
pred.label
head(pred.label)
table(test_y,pred.label)
sum(diag(table(test.y,pred.label)))/1000
```