Introduction

Name: Lee Boon Tat

Title: MEAN Stack Developer / Software Application Support

Summary: Experience Software Application Support Specialist | MEAN Stack developer

-Expertise crafting dynamic user interfaces and implementing scalable server-side solution as MEAN stack developer proficiency in MongoDB, Express.js, Angular, and Node.js, architecting web application from concept to deployment. Previous role as Software Application support role acquire problem-solving abilities and customer centric approach to support delivery using MSSQL and TSQL tools. Commitment to continuous learning and staying updated on emerging technologies with best practices is my personal quotes.

Skills Technologies

Web Technologies: R, Matlab

Databases: MSSQL, Excel Duckascopy

Tools & Framework: R Studio

Operation Systems: Windows

Project Title

Forex Recommendation System

This project going to identify and use a suitable machine learning techniques for FOREX trading patterns modelling using historical data with two currency pair USD/EUR and USD/JYP from January 2004 to December 2013. This 10 years historical data contains daily interval with Open, Close, High and Low prices. The scope of this project focus on Machine Learning comparison within two widely use Machine Learning techniques, which is Artificial Neural Networks (ANN) and Random Forest for modelling the FOREX trends patterns for providing the recommendation on FOREX trading (buy or sell). This project also will adopt one of the commonly use trading rules for analysis and design the recommendation system. The outcome of this project allow new traders using machine learning model for FOREX prediction.

Code Decumentation

Data Preprocessing

Downloaded historical data content 10 years data for USD/JYP and EUR/USD from the date of January 2004 to December 2013 under Figure 3.1 data collection stage. First, we load the necessary libraries in RStudio and then read the EUR/USD data.

```
forex <- read.csv("C:\\forex_data\\USDJYP.csv", header =TRUE)
c<-forex[,5]
o<-forex[,2]
h<-forex[,3]
l<-forex[,4]
v <- forex[,6]
DataFrame <- data.frame(o,h,l,c,v)


#install.packages("neuralnet")
library(neuralnet)
```

Figure 3.2.1 **Import library and data**

Within the figure 3.2.1 data observation, we found out that the data set have no missing value for both pair currency that explained downloaded daily historical dataset USD/JYP and USD/EUR.

```
> apply(forex, 2, function(x) sum(is.na(x)))
Local.time       Open       High        Low      Close     Volume
         0          0          0          0          0          0
```

Figure 3.2.1 **Pre-processing missing value**

```
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x)))
}
maxmindf <- as.data.frame(lapply(DataFrame, normalize))
```

Figure 3.2.3 **Nomalization function of data frame**

Train and Validate Model ANN

Therefore, we use close variable as dependant/target variable for both modelling to predict the closed price value for next day. Below figure 3.4.1 Random sample size with splitting training 70% and testing 30% for the ANN modelling of USD/JYP and EUR/USD forex historical data set.

```
ind <- sample(1:nrow(maxmindf),round(0.70*nrow(maxmindf)))
trainDF <- maxmindf[ind,]
testDF <- maxmindf[-ind,]
```

Figure 3.4.1 **ANN splitting data USD/JYP and EUR/USD**

```
allVars <- colnames(maxmindf)
predictorVars <- allVars[!allVars%in%"c"]
predictorVars <- paste(predictorVars, collapse = "+")
form <- as.formula(paste("c~", predictorVars, collapse = "+"))

neuralModel <- neuralnet(formula = form, hidden = c(5,3), linear.output = TRUE, data = trainDF, threshold=0.01)
```

Figure 3.4.2 **ANN model formula for both pair currency**

Therefore, figure 3.4.8 calculating the average error we use loop for neural network cross validation with cv.glm() function with boot package for 10 K-fold validation.

```
> cv.glm(DataFrame,lm.fit,K=10)$delta[1]
[1] 0.05022452
```

Figure 3.4.8 ANN validation function result

Train and Validate Model RF

Figure 3.5.1 work on creating feature and target for setting up the x-axis and y-axis before training the model by using dplyr packages. Y-axis as close variable and x-axis as predictor variable. We train the model of 400 random sample (reduce the outcome time) with data partition of 70% training data and 30% test data showed in Figure 3.5.2.

```
# Create features and target
library(dplyr)
X <- maxmindf %>%
  select(o,h,l,v)
y <- maxmindf$c
```

Figure 3.5.1 random forest create feature and target

```
####create train and test data set
ind <- sample(1:nrow(maxmindf),400)

##install.packages("caret")
library(caret)
# Split data into training and test sets
index <- createDataPartition(y, p=0.7, list=FALSE)
X_train <- X[ index, ]
X_test <- X[-index, ]
y_train <- y[index]
y_test<-y[-index]
```

Figure 3.5.2 Random forest splitting data

```
# Make prediction
predictions <- predict(regr, X_test)

result <- X_test
result['CloseActualValues'] <- y_test
result['prediction']<-  predictions
```

Figure 3.5.3 random forest prediction and testing formula

```
# Set grid search parameters and Cross Validation with Manual Fine Tuning
control <- trainControl(method="repeatedcv", number=10, repeats=3, search='grid')

# Outline the grid of parameters
tunegrid <- expand.grid(.maxnodes=c(70,80,90,100), .ntree=c(900, 1000, 1100))
set.seed(seed)
```

Figure 3.5.8 Random forest cross validation 10 k-fold with manual fine tuning

Results

Overall, RF performed better than ANN for predicting close price with Mean Square Error of 0.0017(accuracy=99.84%) and 0.065(accuracy=99.99%) respectively in USD/JYP. ANN performed better than RF for predicting close price with Mean Square Error of 0.000008(accuracy=99.98%) and 0.0012(accuracy=99.95%) respectively in EUR/USD. RF performs internal cross-validation and is relatively easy to tune as it has few tuning parameters. Cross validation testing used in ANN model are 0.0000009 and 0.075 respectively. The random forest model is the better model compare to artificial neural network when come to predicting forex using machine learning algorithms linear model for generate better recommendation system.