National College of Ireland

# Configuration Manual

MSc Research Project
Fintech

Yen Lyn Ooi
Student ID: X19128657

School of Computing
National College of Ireland

Supervisor: Victor Del Rosal

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

| | |
|---|---|
| **Student Name:** | Yen Lyn Ooi |
| **Student ID:** | X19128657 |
| **Programme:** | MSc FinTech **Year:** 2020 |
| **Module:** | Research Project |
| **Lecturer:** | Victor Del Rosal |
| **Submission Due Date:** | 17th August 2020 |
| **Project Title:** | Analysis of Cryptocurrency Public Sentiment Shifts |
| **Word Count:** | 855 **Page Count:** 8 |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** .....

**Date:** .....

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies) | □ |
| **Attach a Moodle submission receipt of the online project submission,** to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Configuration Manual

Yen Lyn Ooi
Student ID: X19128657

## 1 Introduction

This configuration manual describes how configure the text mining application with the hardware and software requirements. The guide provides the steps to run the codes in related to the research titled "*How has sentiment towards cryptocurrencies shifted over the years as evidenced in public media and news?"*

## 2 Hardware Requirements

Laptop Model: Dell Inspiron 5468

Processor: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.70 GHz

Installed Memory (RAM): 4 GB

GPU: Intel (R) HD Graphics 620

System: 64-bit operating system, x64-based processor

## 3 Software Requirements

R version: 3.6.2

R studio version: 1.2.5033

Operating System: Windows 10

Microsoft Excel Version: 2002

## 4 Dataset

The dataset is obtained from Kaggle webpage, https://www.kaggle.com/kashnitsky/news-about-major-cryptocurrencies-20132018-40k. The dataset consists of cryptocurrency information collected from five lists of media: Cointelegraph, News BTC, CoinDesk, CCN and ForkLog in the period of

2013 to 2017. There are totals 39,467 instances and 7 attributes included in the dataset The seven attributes consist of the URL of five different media news, title of cryptocurrency news, the text of the news, hypertext markup language (HTML), year, the news author and sources of news such as interview, news and opinions.

# 5 Code

Install needed packages and load the libraries for sentiment analysis.

```
install.packages('tidytext')
install.packages('tm')
install.packages("SnowballC")
install.packages("wordcloud")
install.packages("RColorBrew")
install.packages("textdata")

library(dplyr)
library(tidytext)
library(textdata)
library(vctrs)
library(tidyr)
library(tm)
library(SnowballC)
library(wordcloud)
library(RColorBrewer)
library(ggplot2)
library(gridExtra)
library(e1071)
library(caret)
```

Load data files into R.

```
Crypto1=read.csv(file="cryptonews1.csv",header= TRUE,
stringsAsFactors = FALSE)
Crypto2=read.csv(file="cryptonews2.csv",header= TRUE,
stringsAsFactors = FALSE)
```

Merge two datasets into one and subset the needed variables.

```
merge_crypto=rbind(Crypto1, Crypto2)
subset=merge_crypto[c('title','year')]
```

First create corpus for the data. Thereafter, conduct text cleaning with the following steps.

```
cryptonews= Corpus(VectorSource(subset))
toSpace= content_transformer(function(x, pattern)
gsub(pattern, " ", x))
cryptonews2= tm_map(cryptonews, toSpace, "â˜")
cryptonews2= tm_map(cryptonews, content_transformer(tolower))
cryptonews2= tm_map(cryptonews, removeNumbers)
```

```
cryptonews2= tm_map(cryptonews, removeWords,
stopwords("english"))
cryptonews2= tm_map(cryptonews, removeWords, c("will","can"))
cryptonews2= tm_map(cryptonews, stripWhitespace)
cryptonews2= tm_map(cryptonews, removePunctuation)
cryptonews2= tm_map(cryptonews, stemDocument)

inspect(cryptonews[["1"]])
inspect(cryptonews2[["1"]])
```

Build a term-document matrix for data exploration with the most frequent words and comparison word cloud.

```
dtm = TermDocumentMatrix(cryptonews2)
dtm0= as.matrix(dtm)
dtm_v <- sort(rowSums(dtm0),decreasing=TRUE)
dtm_d <- data.frame(word = names(dtm_v),freq=dtm_v)

head(dtm_d, 5)
barplot(dtm_d[1:50,]$freq, las = 2, names.arg =
dtm_d[1:50,]$word,
        col = rainbow(10),main ="Top 50 most frequent words",
        ylab = "Word frequencies")
.
colnames(dtm0)= c(2013:2018)
comparison.cloud(dtm0[,1:6], random.order=T
                 ,max.words=100
                 ,min.freq=5
                 ,scale=c(2,0.5),title.size=1.5,
                 ,colors=rainbow(12)
)
```

Next, tidy text format is created to perform sentiment analysis with Bing lexicon.

```
crypto_tidy <- subset2 %>%
unnest_tokens(word, title) %>%
filter(!nchar(word) < 3) %>%
anti_join(stop_words)
glimpse(crypto_tidy)

yr_count= subset2 %>% group_by(year) %>% summarise(count=n())
plot(yr_count,type = "o",col = "red", xlab = "Year", ylab =
"n",
     main = "Distribution of Crypto Headlines")

c_bing <- crypto_tidy %>%
  inner_join(get_sentiments("bing"))

bing_plot <- c_bing %>%
  group_by(sentiment) %>%
```

```
  summarise(word_count = n()) %>%
  ungroup() %>%
  mutate(sentiment = reorder(sentiment, word_count)) %>%
  ggplot(aes(sentiment, word_count, fill = -word_count)) +
  geom_col() +
  labs(x = NULL, y = "Word Count") +
  scale_y_continuous(limits = c(0, 15000)) +
  ggtitle("Crypto Bing Sentiment") +
  coord_flip()

bing_plot
```

Apply TD-IDF function to find the commonly used words in the content by reducing the weight for commonly used words and increasing the weight for words that are not used very much in a collection or corpus of documents. The second model is created with TD-IDF for Bing sentiment.

```
c_tf_idf <- crypto_tidy %>%
  count(word, year) %>%
  bind_tf_idf(word,year, n) %>%
  arrange(desc(tf_idf))
c_tf_idf

c_bing2 <- c_tf_idf %>%
  inner_join(get_sentiments("bing"))

bing_plot2 <- c_bing2 %>%
  group_by(sentiment) %>%
  summarise(word_count = n()) %>%
  ungroup() %>%
  mutate(sentiment = reorder(sentiment, word_count)) %>%
  ggplot(aes(sentiment, word_count, fill = -word_count)) +
  geom_col() +
  labs(x = NULL, y = "Word Count") +
  scale_y_continuous(limits = c(0, 15000)) +
  ggtitle("Crypto_tf_idf Sentiment") +
  coord_flip()

bing_plot2

com <- c_bingn %>%
  filter(n>150) %>%
  mutate(n = ifelse(sentiment == "negative", -n, n)) %>%
  mutate(word = reorder(word, n))

ggplot(data = com, mapping = aes(x = word, y = n, fill =
sentiment)) +
  geom_bar(alpha = 1, stat = "identity") +
  labs(y = "Contribution to sentiment", x = NULL) +
  ggtitle("The most common positive and negative words") +
  coord_flip()
```

The following codes is used to calculate the polarity score.

```
c_polarity_year <- c_bing %>%
  count(sentiment, year) %>%
  spread(sentiment,n, fill = 0) %>%
  mutate(polarity = (positive - negative)/ (positive +
negative) * 100,
         pos.polarity = positive / (positive + negative) *
100)

c_polarity <- c_bing %>%
  count(sentiment, word) %>%
  spread(sentiment,n,fill=0) %>%
  mutate(polarity = (positive - negative) / (positive +
negative))

c_over_time <- c_polarity_year %>%
  ggplot(aes(year, polarity, color = ('red'))) +
  geom_col() +
  theme(legend.position="none") +
  geom_smooth(method = "lm", se = FALSE, aes(color = 'blue'))
+
  guides(fill = FALSE)+
  theme(plot.title = element_text(size = 11)) +
  xlab(NULL) + ylab(NULL) +
  ggtitle("Polarity Over 5 Years Time")

relative_polarity_over_time <- c_polarity_year %>%
  ggplot(aes(year, pos.polarity , color=('red'))) +
  geom_col() + theme(legend.position="none")+
  geom_smooth(method = "lm", se = FALSE, aes(color = 'blue'))
+
  guides(fill = FALSE) +
  theme(plot.title = element_text(size = 11)) +
  xlab(NULL) + ylab(NULL) +
  ggtitle("Positive sentiment Over 5 Years Time")

grid.arrange(c_over_time, relative_polarity_over_time, ncol =
2)
```

The last step is to apply SVM with the polarity score to train the general lexicon model and TD-IDF model.

```
set.seed(1)
index = sample(1:nrow(c_bing),size=nrow(c_bing)*0.80,
replace=FALSE)
train_Csub=c_bing[index,]
test_Csub=c_bing[-index,]
```

```
svm_model= svm(sentiment ~ word, kernel= "linear",
data=train_Csub, cost= 10, scale=FALSE, type='C')
summary(svm_model)

predsvm=predict(svm_model,test_Csub)
table(predsvm,test_Csub$sentiment)
agreement=predsvm == test_Csub$sentiment
prop.table(table(agreement))

confusionMatrix(predsvm,reference= test_Csub$sentiment)

set.seed(3)
index = sample(1:nrow(c_bing2),size=nrow(c_bing2)*0.80,
replace=FALSE)
train_Csub2=c_bing2[index,]
test_Csub2=c_bing2[-index,]

svm_model2= svm(sentiment ~ year, kernel= "linear",
data=train_Csub2, cost= 10, scale=FALSE, type='C')
summary(svm_model2)

predsvm2=predict(svm_model2,test_Csub2)
table(predsvm2,test_Csub2$sentiment)
agreement2=predsvm2 == test_Csub2$sentiment
prop.table(table(agreement2))

confusionMatrix(predsvm2,reference= test_Csub2$sentiment)
```