Diebold

Libraries required

```
library(readxl)
library(dplyr)
library(magrittr)
library(tm)
library(keras)
library(caret)
library(SnowballC)
library(reshape2)
```

Loading the data

Trying to encode specific types of values to NA

```
diebold_non_billed <- read_excel("december_non-bill_calls.xlsx", na = c("", "---"), col_
types = "text")
diebold_billed <- read_excel("december_billed_calls.xlsx", na = c("", "---"), col_types
 = "text")
```

Data Preparation

Removing the variables that were not important, based on the initial discussions and research into the data set.

```
unimportant_variables <- c("SR Address Line 1", "SR City", "SR Status", "Activity Statu
s", "Charges Status", "SR Coverage Hours...11", "SR Coverage Hours...28","SR Contact Dat
e","Br Area Desc", "Activity Facts Call Num","Activity Completed Date", "Item Desc", "SR
Site","SR Serial Number", "Base Call YN", "Br Region Desc", "SR Number","Br Branch Desc"
)
unimportant_variables
```

```
##  [1] "SR Address Line 1"       "SR City"
##  [3] "SR Status"               "Activity Status"
##  [5] "Charges Status"          "SR Coverage Hours...11"
##  [7] "SR Coverage Hours...28"  "SR Contact Date"
##  [9] "Br Area Desc"            "Activity Facts Call Num"
## [11] "Activity Completed Date" "Item Desc"
## [13] "SR Site"                 "SR Serial Number"
## [15] "Base Call YN"            "Br Region Desc"
## [17] "SR Number"               "Br Branch Desc"
```

```
diebold_non_billed <- diebold_non_billed %>% select(-c(unimportant_variables))
diebold_billed <- diebold_billed %>% select(-c(unimportant_variables))
```

Combining both the billed and non billed diebold data into a single data frame.

```
diebold_df <- bind_rows(diebold_non_billed, diebold_billed)
```

Assigning the value “0” to NA values

```
diebold_df[is.na(diebold_df)] <- 0
```

Data Cleaning

Encoding the categorical variables as factors

```
## [1] "Invoiced (Y/N)"
## [2] "Activity Type"
## [3] "Activity Trouble Code"
## [4] "Coverage Type"
## [5] "SR Type"
## [6] "SR Device"
## [7] "SR Owner (Q#)"
## [8] "Cash Vendor & Consumable Contracts"
## [9] "SR State"
```

```
diebold_df <- diebold_df %>% mutate_at(characters, factor)
summary(diebold_df)
```

```
##  Invoiced (Y/N)        SR Owner (Q#)   Billing Notes
##  N:114318       FS5-0000000001:26977   Length:141279
##  Y: 26961       FS5-REGIONAL  :14423   Class :character
##                 FS5-0000000008:13940   Mode  :character
##                 FS5-0000000010:11008
##                 FS5-0000000135:10822
##                 FS5-0000000015: 9413
##                 (Other)       :54696
##   Call Text                     Cash Vendor & Consumable Contracts
##  Length:141279      0                              :135409
##  Class :character   Consumables Contract           :  1669
##  Mode  :character   Vendor & Consumable Contracts:  1272
##                     Vendor Contract                :  2929
##
##
##
##                  SR Type
##  FL - First Line Call:94565
##  IN - BW             : 1294
##  IN - SO             :    4
##  PM/CL               :   26
##  SW                  :  118
##  TR - Trouble Call   :45272
##
##                                                Coverage Type   SR Device
##  FIRST LINE OPTEVA 760 - MANAGED PLAN               :13400   ALM:     2
##  GO PLAN:  Gold Coverage - Conventional & ATM Produ:12820   CAM:     2
##  FIRST LINE - ENA ENHANCED NOTE ACCEPTOR - MANAGED :12370   MSC:   154
##  0                                                  :11150   POS:     8
##  FIRST LINE OPTEVA 750 - MANAGED PLAN               : 8897   TAB:124405

##  BR PLAN:  Bronze Coverage - ATM Products           : 8057   VAT:  4713
##  (Other)                                            :74585   VLT: 11995
##                 Activity Trouble Code    SR State
##  REPAIR                      :50691     CA     :16240
##  JM_JAMMED                   :23532     FL     :11226
##  FL_FIRSTLINE                :19544     NY     : 8642
##  BC_BANK CUSTOMER ERROR      : 9989     PA     : 8258
##  CASH ERROR_VENDOR_CUSTOMER: 8039     TX     : 8179
##  XX_NONE OF THE ABOVE        : 5551     ON     : 6517
##  (Other)                     :23933     (Other):82217
##             Activity Type
##  First Line         :94309
##  Field Repair       :44999
##  Installation - BW : 1294
##  Opteview           :  267
##  Forced Entry       :  189
##  SW Trouble/Upgrade:  118
##  (Other)            :  103
```

“Billing Notes” and “Call Text” are the two variables in our data set that are free Form Text

```
diebold_call_text <-  use_series(diebold_df, `Call Text`)
diebold_billing_notes <-  use_series(diebold_df, `Billing Notes`)
```

```
call_text_corpus <- VCorpus(VectorSource(diebold_call_text), readerControl = list(langua
ge = "en"))
bill_notes_corpus <- VCorpus(VectorSource(diebold_billing_notes), readerControl = list(l
anguage = "en"))
```

Free form text - Data Cleaning

```
replace_asterix <- function(document) {gsub(pattern = "\\*", replacement = " ", documen
t)}
add_space_period <- function(document) {gsub(pattern = "\\.", replacement = ". ", docume
nt)}
remove_single_chars <- function(document) {gsub(pattern = "\\s[a-z]\\s", replacement = "
", document)}
clean_text <- function(corpus) {corpus %>% tm_map(content_transformer(tolower)) %>% tm_m
ap(content_transformer(replace_asterix)) %>% tm_map(content_transformer(add_space_perio
d)) %>% tm_map(removeNumbers) %>% tm_map(removeWords, stopwords("english")) %>% tm_map(r
emoveWords, c("pm", "am", "edt")) %>% tm_map(removePunctuation) %>% tm_map(content_trans
former(remove_single_chars)) %>% tm_map(stripWhitespace) %>% tm_map(content_transformer
(trimws)) %>% tm_map(stemDocument)}
call_text_cleaned <- clean_text(call_text_corpus)
bill_notes_cleaned <- clean_text(bill_notes_corpus)
```

```
diebold_df$`Call Text` <- call_text_cleaned %>% sapply(function (doc) doc$content)
diebold_df$`Billing Notes` <- bill_notes_cleaned %>% sapply(function (doc) doc$content)
```

Target variable - Data preparation

```
invoiced <- diebold_df %>%
  use_series("Invoiced (Y/N)") %>%
  as.numeric() %>%
  subtract(1) %>%
  as.matrix()
dim(invoiced)
```

```
## [1] 141279      1
```

Tokenization

```
CONSTANTS <- list(
  MAX_WORDS = 20000,
  MAX_LEN = 200
)
```

Tokeninzing Categorical data

```
categ_to_tokenize <- c("Invoiced (Y/N)", "Activity Type", "Activity Trouble Code", "Cove
rage Type", "SR Type", "SR Device", "SR Owner (Q#)", "Cash Vendor & Consumable Contract
s", "SR State")
categoricals <- diebold_df %>%select(categ_to_tokenize[-1]) %>% mutate_all(addNA)
```

Dimension

```
categorical_model <- dummyVars(" ~ .", data = categoricals, fullRank = T)
categorical_data <- data.matrix(predict(categorical_model, newdata = categoricals))
dim(categorical_data)
```

```
## [1] 141279    337
```

Tokenizing the free Form Text

Call Text

```
call_text <- diebold_df %>% select(c("Call Text"))
tokenizer <- text_tokenizer(num_words = CONSTANTS$MAX_WORDS) %>%
fit_text_tokenizer(call_text$`Call Text`)
sequences <- texts_to_sequences(tokenizer, call_text$`Call Text`)
```

Unique tokens

```
word_index <- tokenizer$word_index
 length(word_index)
```

```
## [1] 66802
```

Dimension

```
free_form_call_text <- pad_sequences(sequences, maxlen = CONSTANTS$MAX_LEN)
 dim(free_form_call_text)
```

```
## [1] 141279    200
```

Billing Notes

```
billing_notes <- diebold_df %>% select("Billing Notes")
tokenizer <- text_tokenizer(num_words = CONSTANTS$MAX_WORDS) %>%
fit_text_tokenizer(billing_notes$`Billing Notes`)
sequences <- texts_to_sequences(tokenizer, billing_notes$`Billing Notes`)
```

Unique tokens

```
word_index <- tokenizer$word_index
 length(word_index)
```

```
## [1] 2790
```

Dimension

```
free_form_billing_notes <- pad_sequences(sequences, maxlen = CONSTANTS$MAX_LEN)
 dim(free_form_billing_notes)
```

```
## [1] 141279    200
```