Diebold Test

Libraries required

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

Loading the test data

Trying to encode specific types of values to NA

```
diebold_test<- read_excel("February_Test_Date_Corrected.xlsx", na = c("", "---"), col_ty
pes = "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...29","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", "Tech Name", "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...29"  "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] "Tech Name"               "SR Number"
## [19] "Br Branch Desc"
```

```
diebold_test <- diebold_test %>% select(-c(unimportant_variables))
```

Assigning the value “0” to NA values

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

```
colnames(diebold_test)
```

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

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_test <- diebold_test %>% mutate_at(characters_test, factor)
```

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

```
diebold_test_call_text <-  use_series(diebold_test, `Call Text`)
diebold_test_billing_notes <-  use_series(diebold_test, `Billing Notes`)
```

```
diebold_test_call_text_corpus <- VCorpus(VectorSource(diebold_test_call_text), readerCon
trol = list(language = "en"))
diebold_test_bill_notes_corpus <- VCorpus(VectorSource(diebold_test_billing_notes), read
erControl = list(language = "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)}
diebold_test_call_text_cleaned <- clean_text(diebold_test_call_text_corpus)
diebold_test_bill_notes_cleaned <- clean_text(diebold_test_bill_notes_corpus)
```

```
diebold_test$`Call Text` <- diebold_test_call_text_cleaned %>% sapply(function (doc) doc
$content)
diebold_test$`Billing Notes` <- diebold_test_bill_notes_cleaned %>% sapply(function (do
c) doc$content)
```

Target variable - Data preparation

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

```
## [1] 108873      1
```

Tokenization

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

Tokeninzing Categorical data

Tokeninzing Categorical data

```
categ_to_tokenize_test <- c("Invoiced (Y/N)", "Activity Type", "Activity Trouble Code",
"Coverage Type", "SR Type", "SR Device", "SR Owner (Q#)", "Cash Vendor & Consumable Cont
racts", "SR State")
categoricals_test <- diebold_test %>%select(categ_to_tokenize_test[-1]) %>% mutate_all(a
ddNA)
```

Dimension

```
categorical_model_test <- dummyVars(" ~ .", data = categoricals_test, fullRank = T)
categorical_data_test <- data.matrix(predict(categorical_model_test, newdata = categoric
als_test))
padded_categorical <- cbind(categorical_data_test, matrix(0, nrow = 108873, ncol = 19))
dim(padded_categorical)
```

```
## [1] 108873    337
```

Tokenizing the free Form Text

Call Text

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

Unique tokens

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

```
## [1] 62385
```

Dimension

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

```
## [1] 108873    200
```

Billing Notes

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

Unique tokens

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

```
## [1] 719
```

dimension

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

```
## [1] 108873    200
```

Training Model on February Test Data

```
probability_feb_test <- predict(model, list(test_free_form_billing_notes, test_free_form
_call_text,padded_categorical),batch_size = 128)
class_prediction_feb <- as.numeric(probability_feb_test >= .30) %>% as.factor() %>% as.d
ata.frame()
colnames(class_prediction_feb) <- c("Invoiced (Y/N)")
write.csv(class_prediction_feb, file = "Diebold_prediction_file.csv", row.names = F)
```