edgar\_partb

Gibran Makyanie

27/04/2020

# Data Preparation

conn <- dbConnect(RSQLite::SQLite(), "edgar.db")

# Text Feature Extraction

## Importing Louhran & McDonald’s stopwords

# -- Import LM Dictionary  
LM\_dictionary\_file <- "LoughranMcDonald\_SentimentWordLists\_2018.xlsx"  
sentiment <- c("negative", "positive", "uncertainty", "litigious", "strong\_modal","weak\_modal", "constraining")  
   
LM\_dictionary <- data.frame()  
for(s in 1:7) {  
 local\_df <- tibble(word = tolower(read\_excel(LM\_dictionary\_file, sheet = s+1)$word), sentiment = sentiment[s] )  
 LM\_dictionary <- bind\_rows(LM\_dictionary, local\_df)  
}  
  
rm(local\_df)  
  
dummy\_LM <- tibble(accession\_number = 'dummy', positive = 0, negative = 0, uncertainty = 0, litigious = 0, constraining = 0, strong\_modal = 0, weak\_modal = 0)  
dummy\_nrc <- tibble(accession\_number = 'dummy', positive = 0, negative = 0, anger = 0, fear = 0, trust = 0, sadness = 0, surprise = 0, disgust = 0, joy = 0, anticipation = 0)  
dummy\_bing <- tibble(accession\_number = 'dummy', positive = 0, negative = 0)

## Calculating Sentiment Measures and Loading to sentiment table

cik <- dbGetQuery(conn, 'SELECT distinct(cik) as cik FROM master\_index')$cik  
  
  
for (c in 1:length(cik)) {  
 df\_cik <- dbGetQuery(conn, paste0('SELECT accession\_number, cleaned\_text FROM master\_index WHERE cik = "',cik[c], '"'))  
   
 for (r in 1:nrow(df\_cik)) {  
   
 df\_report <- dbGetQuery(conn, paste0('SELECT accession\_number, cleaned\_text FROM master\_index WHERE accession\_number = "',df\_cik$accession\_number[r], '"'))  
 # ----- Sentiment Syuzhet, Vader, and n\_words  
 sentiment\_syuzhet\_vader <- textfeatures(df\_report$cleaned\_text, normalize = FALSE, word\_dims = FALSE, sentiment = TRUE) %>%  
 select(sent\_syuzhet,sent\_vader) %>%  
 mutate(accession\_number = df\_report$accession\_number)  
   
 # ----- Text Complexity  
 tokenised <- df\_report %>%   
 unnest\_tokens(word, cleaned\_text)  
   
 n\_words <- tokenised %>%  
 group\_by(accession\_number) %>%  
 count(accession\_number)  
   
 n\_complex <- tokenised %>%  
 group\_by(word, accession\_number) %>%  
 mutate(complexity = nchar( gsub( "[^X]", "", gsub( "[aeiouy]+", "X", tolower( word ))))) %>%  
 filter(complexity >=3) %>%  
 group\_by(accession\_number) %>%  
 count(accession\_number)  
   
 complexity <- tibble(accession\_number = n\_words$accession\_number,  
 complexity = n\_complex$n / n\_words$n)  
   
 rm(n\_complex)  
   
 # ----- Sentiment LoughranMcDonald  
 tokens\_LM <- tokenised %>%  
 inner\_join(LM\_dictionary)   
   
 word\_count\_LM <- tokens\_LM %>% group\_by(accession\_number) %>% summarise(LM\_total\_words =n())  
   
 sentiment\_LM <- tokens\_LM %>%   
 group\_by(accession\_number,sentiment) %>%   
 summarise(total\_sentiment = n()) %>%   
 spread(sentiment, total\_sentiment, fill = 0) %>%  
 bind\_rows(dummy\_LM) %>%  
 left\_join(word\_count\_LM) %>%  
 mutate(LM\_sent = positive - negative,  
 LM\_positive = positive / LM\_total\_words,  
 LM\_negative = negative / LM\_total\_words,  
 LM\_uncertainty = uncertainty / LM\_total\_words,  
 LM\_litigious = litigious / LM\_total\_words,  
 LM\_constraining = constraining / LM\_total\_words,  
 LM\_strong\_modal = strong\_modal / LM\_total\_words,  
 LM\_weak\_modal = weak\_modal / LM\_total\_words) %>%  
 select(-c(positive, negative, uncertainty, litigious, constraining, strong\_modal, weak\_modal)) %>%  
 filter(accession\_number != 'dummy')  
   
 rm(tokens\_LM, word\_count\_LM)  
   
 # ----- Sentimentr  
 text <- get\_sentences(df\_report$cleaned\_text)  
 sentimentr <- tibble(accession\_number = df\_report$accession\_number,   
 sentimentr = as.data.frame(sentiment\_by(text))$ave\_sentiment)  
   
 rm(text)  
   
 # ----- Sentiment Afinn  
 sentiment\_afinn <- tokenised %>%  
 inner\_join(get\_sentiments("afinn")) %>%  
 group\_by(accession\_number) %>%   
 summarise(afinn\_sent = sum(value))  
   
 # ----- Sentiment bing  
 tokens\_bing <- tokenised %>%   
 inner\_join(get\_sentiments("bing"))  
   
 word\_count\_bing <- tokens\_bing %>% group\_by(accession\_number) %>% summarise(bing\_total\_words =n())  
   
 sentiment\_bing <- tokens\_bing %>%   
 group\_by(accession\_number,sentiment) %>%   
 summarise(total\_sentiment = n()) %>%   
 spread(sentiment, total\_sentiment, fill = 0) %>%  
 bind\_rows(dummy\_bing) %>%  
 left\_join(word\_count\_bing) %>%  
 mutate(bing\_sent = positive - negative,  
 bing\_positive = positive/bing\_total\_words,  
 bing\_negative = negative/bing\_total\_words) %>%  
 select(-c(positive, negative)) %>%  
 filter(accession\_number != 'dummy')  
   
 rm(tokens\_bing, word\_count\_bing)  
   
 # ----- Sentiment NRC  
 tokens\_nrc <- tokenised %>%   
 inner\_join(get\_sentiments("nrc"))   
   
 word\_count\_nrc <- tokens\_nrc %>% group\_by(accession\_number) %>% summarise(nrc\_total\_words =n())  
   
 sentiment\_nrc <- tokens\_nrc %>%   
 group\_by(accession\_number,sentiment) %>%   
 summarise(total\_sentiment = n()) %>%   
 spread(sentiment, total\_sentiment, fill = 0) %>%  
 bind\_rows(dummy\_nrc) %>%  
 left\_join(word\_count\_nrc) %>%  
 mutate(nrc\_sent = positive - negative,  
 nrc\_positive = positive / nrc\_total\_words,  
 nrc\_negative = negative / nrc\_total\_words,  
 nrc\_anger = anger/nrc\_total\_words,  
 nrc\_fear = fear/nrc\_total\_words,  
 nrc\_trust = trust/nrc\_total\_words,  
 nrc\_sadness = sadness/nrc\_total\_words,  
 nrc\_surprise = surprise/nrc\_total\_words,  
 nrc\_disgust = disgust/nrc\_total\_words,  
 nrc\_joy = joy/nrc\_total\_words,  
 nrc\_anticipation = anticipation/nrc\_total\_words) %>%  
 select(-c(positive, negative, anger, trust, sadness, surprise, disgust,joy, anticipation, fear )) %>%  
 filter(accession\_number != 'dummy')  
   
 rm(tokens\_nrc, word\_count\_nrc)  
   
 # ----- Merging Sentiment Features  
 sentiment\_df <- sentiment\_LM %>%  
 left\_join(complexity) %>%  
 left\_join(sentimentr) %>%  
 left\_join(sentiment\_afinn) %>%  
 left\_join(sentiment\_bing) %>%  
 left\_join(sentiment\_nrc) %>%  
 left\_join(sentiment\_syuzhet\_vader)  
   
 rm(sentiment\_LM, complexity, sentimentr, sentiment\_afinn, sentiment\_bing, sentiment\_nrc, sentiment\_syuzhet\_vader, n\_words, tokenised)  
  
 # ----- Insertion to SQL table  
 dbWriteTable(conn,"sentiment", sentiment\_df, append = TRUE) # insert to sentiment\_df Table  
   
 rm(sentiment\_df)  
  
 }  
   
 print(paste(c, "of", length(cik), "sentiment calculated"))  
}  
  
rm(df\_cik, df\_report, dummy\_bing, dummy\_LM, dummy\_nrc, LM\_dictionary, cik, c, r, sentiment, LM\_dictionary\_file)

## Downloading Stock Price

# ----- Set up columns  
dbExecute (conn, 'ALTER TABLE master\_index ADD COLUMN return\_adjusted\_price double;')  
dbExecute (conn, 'ALTER TABLE master\_index ADD COLUMN price\_adjusted\_ratio double;')

# ----- Downloading and Loading to master\_index table  
cik <- dbGetQuery(conn, 'SELECT distinct(cik) as cik FROM master\_index')$cik  
  
for (c in 1:length(cik)) {  
   
 # ----- Import per CIK   
 df\_cik <- dbGetQuery(conn, paste0('SELECT master\_index.cik, form\_type, date\_filed, accession\_number, symbol FROM master\_index   
 LEFT JOIN (SELECT cik, symbol FROM sp500 group by cik) AS sp500  
 ON master\_index.cik = sp500.cik  
 WHERE master\_index.cik = "', cik[c] ,'"')) %>% mutate(date\_filed = as.Date(date\_filed, origin="1970-01-01"))  
   
   
 for (r in 1:nrow(df\_cik)) {  
 tryCatch({  
 df\_report <- df\_cik[r,] # iterate for every row  
   
 # ----- Get stock information  
 stock\_data <- BatchGetSymbols(tickers = df\_report$symbol,  
 first.date= df\_report$date\_filed - 7,  
 last.date= df\_report$date\_filed + 3,  
 type.return="log")  
   
 # ----- Filter the 2nd day and the last day  
 stock\_data\_filtered <- stock\_data[[2]] %>%  
 filter(ref.date == max(ref.date) | row\_number() == 2) %>%   
 arrange(desc(ref.date))  
   
 # ----- Calculate stock price change on log scale  
 return\_adjusted\_price <- stock\_data\_filtered$ret.closing.prices[1] - stock\_data\_filtered$ret.closing.prices[2] # return difference  
 price\_adjusted\_ratio <- (stock\_data\_filtered$price.adjusted[1] / stock\_data\_filtered$price.adjusted[2]) - 1 # stock price ratio  
   
 accession\_number <- df\_report$accession\_number[1]  
   
 # ----- Update to DB  
 dbExecute(conn, paste0("UPDATE master\_index SET return\_adjusted\_price = ",return\_adjusted\_price ,", price\_adjusted\_ratio = ",price\_adjusted\_ratio ,"  
 WHERE accession\_number = '",accession\_number ,"'"))  
   
   
 print(paste(accession\_number, "has been processed..."))  
   
 }, error=function(e){cat("ERROR :",conditionMessage(e), "\n")})  
 }  
   
 print(paste(c, "of", length(cik), "stock price change added"))  
}  
  
  
t2 <- Sys.time()  
t2-t1

# Predicting Stock Price Change

## Exploratory Data Analysis

# ----- Data Prep  
sentiment\_data <- dbGetQuery(conn, 'SELECT sentiment.\*, gics\_sector, master\_index.cik, company\_name, year\_filed, form\_type, return\_adjusted\_price, price\_adjusted\_ratio FROM sentiment   
 INNER JOIN master\_index ON sentiment.accession\_number = master\_index.accession\_number   
 INNER JOIN (SELECT \* FROM sp500 group by cik) AS sp500 ON master\_index.cik = sp500.cik  
 ') %>% mutate(year\_filed = as.factor(year\_filed),  
 company\_name = as.factor(company\_name),  
 cik = as.factor(cik),  
 gics\_sector = as.factor(gics\_sector),  
 accession\_number = as.factor(accession\_number),  
 price\_adjusted\_ratio = price\_adjusted\_ratio\*100)  
  
  
  
load('temp/sentiment\_data.rda')  
  
  
  
sentiment\_data <- sentiment\_data %>%   
 filter(LM\_total\_words > 10 & bing\_total\_words > 10, nrc\_total\_words > 10) %>%  
 filter(price\_adjusted\_ratio != !is.na(price\_adjusted\_ratio) | return\_adjusted\_price != is.na(return\_adjusted\_price)) %>%  
 mutate\_if(is.numeric, funs(ifelse(is.na(.), 0, .)))  
  
save(sentiment\_data, file = '/Volumes/Buku Gibran/edgar/temp/sentiment\_data.rda')  
   
plot\_missing(sentiment\_data)  
plot\_histogram(sentiment\_data)

# ----- Plot Average Return  
return\_avg <- sentiment\_data %>%  
 group\_by(gics\_sector, year\_filed, form\_type) %>%  
 summarise(return\_adjusted\_price = mean(return\_adjusted\_price))  
  
return\_avg %>%  
 ggplot(aes(x = year\_filed, y = return\_adjusted\_price, color = gics\_sector)) +  
 geom\_line(aes(group = gics\_sector)) +  
 labs(title = 'Average Log Return between 2009 and 2019 after SEC Filings', subtitle='Grouping on GICS sectors', x = 'Year Filed', y = 'Log Return', legend = 'GICS Sector') + coord\_cartesian(ylim=c(-0.04,0.04)) +  
 facet\_wrap(~form\_type, ncol = 4, scales = "free")  
  
  
# ----- Plot Average Stock Price Change  
price\_avg <- sentiment\_data %>%  
 group\_by(gics\_sector, year\_filed, form\_type) %>%  
 summarise(price\_adjusted\_ratio = mean(price\_adjusted\_ratio))  
  
price\_avg %>%  
 ggplot(aes(x = year\_filed, y = price\_adjusted\_ratio, color = gics\_sector)) +  
 geom\_line(aes(group = gics\_sector)) +  
 labs(title = 'Average Stock Price Change between 2009 and 2019 after SEC Filings', subtitle='Grouping on GICS sectors', x = 'Year Filed', y = '% Change in Stock Price', legend = 'GICS Sector') + coord\_cartesian(ylim=c(-10,10)) +  
 facet\_wrap(~form\_type, ncol = 4, scales = "free")  
  
rm(return\_avg,price\_avg)

## How Sentiment Analysis affect Stock Price Change

# ----- Individual Dictionaries  
load('/Volumes/Buku Gibran/edgar/temp/sentiment\_data.rda')  
  
  
sentiment\_regression\_data <-sentiment\_data %>% filter(form\_type == '10-K') %>% select(-c(form\_type, year\_filed, cik, company\_name, return\_adjusted\_price, gics\_sector, accession\_number))x  
str(sentiment\_regression\_data)  
  
reg\_10k\_LM <- lm(price\_adjusted\_ratio ~ LM\_total\_words + LM\_sent + LM\_positive + LM\_negative + LM\_uncertainty + LM\_litigious + LM\_constraining + LM\_strong\_modal + LM\_weak\_modal + complexity, data = sentiment\_regression\_data)  
reg\_10k\_bing <- lm(price\_adjusted\_ratio ~ bing\_total\_words + bing\_sent + bing\_positive + bing\_negative, data = sentiment\_regression\_data)  
reg\_10k\_sentimentr <- lm(price\_adjusted\_ratio ~ sentimentr, data = sentiment\_regression\_data)  
reg\_10k\_afinn <- lm(price\_adjusted\_ratio ~ afinn\_sent, data = sentiment\_regression\_data)  
reg\_10k\_nrc <- lm(price\_adjusted\_ratio ~ nrc\_sent + nrc\_total\_words + nrc\_positive + nrc\_negative + nrc\_anger + nrc\_fear + nrc\_trust + nrc\_sadness + nrc\_surprise + nrc\_disgust + nrc\_joy + nrc\_anticipation, data = sentiment\_regression\_data)  
reg\_10k\_syuzhet <- lm(price\_adjusted\_ratio ~ sent\_syuzhet, data = sentiment\_regression\_data)  
reg\_10k\_vader <- lm(price\_adjusted\_ratio ~ sent\_vader, data = sentiment\_regression\_data)  
  
library(stargazer)  
  
stargazer::stargazer(reg\_10k\_LM,reg\_10k\_bing,reg\_10k\_sentimentr, reg\_10k\_afinn, reg\_10k\_nrc, reg\_10k\_syuzhet, reg\_10k\_vader, type = "text")

sector <- unique(sentiment\_data$gics\_sector)  
  
  
# ----- Model sentiment for every Sector  
model\_output\_sector <- data.frame()  
  
for (s in 1:length(sector)) {  
 model\_data <- sentiment\_data %>% filter(form\_type == "10-K", gics\_sector == sector[s]) %>%  
 select(-c(accession\_number,year\_filed, form\_type, cik, company\_name, return\_adjusted\_price, gics\_sector))  
   
 # ----- Modelling  
 model <- lm(price\_adjusted\_ratio ~., data = model\_data)  
   
 # ----- Prep data for plotting  
 local\_df <- head(as.data.frame(summary(model)$coefficients) %>%  
 tibble::rownames\_to\_column() %>%  
 mutate(absolute\_t\_value = abs(`t value`)) %>%  
 arrange(desc(absolute\_t\_value)) , 10) %>%  
 mutate(rowname=factor(rowname, levels=rowname)) %>%  
 mutate(significance = case\_when(`Pr(>|t|)` <= 0.001 ~ 'significant\*\*\*', `Pr(>|t|)` <= 0.01 ~ 'significant\*\*', `Pr(>|t|)` <= 0.05 ~ 'significant\*', TRUE ~ 'not significant')) %>%  
 mutate(gics\_sector = sector[s]) %>%  
 mutate(r\_squared = paste('Multiple R-squared:',as.character(round(summary(model)$r.squared,3))))  
   
 model\_output\_sector <- bind\_rows(model\_output\_sector, local\_df)  
}  
  
ggplot(model\_output\_sector, aes(x = reorder(rowname, absolute\_t\_value), y = absolute\_t\_value, fill=significance)) + geom\_bar(stat = "identity") +  
 labs(title ='Top 10 Sentiment Features for predicting Stock Price Change after 10-K Filings', subtitle = 'groupings on GICS Sectors',y = 'Absolute t-value', x = 'Features') +  
 ylim(0,4) + coord\_flip() +  
 facet\_wrap(~gics\_sector+r\_squared, ncol = 4, scales = "free")  
  
rm(model\_output\_sector)

## Modelling 10-K: Building Exhaustive Model to predict Stock Price Change

load(file = '/Volumes/Buku Gibran/edgar/temp/sentiment\_data.rda')  
# ----- Stock Price Modelling  
model\_data\_10k <- sentiment\_data %>%   
 filter(form\_type == "10-K") %>%  
select(-c(form\_type, cik, return\_adjusted\_price))  
  
accession\_number <- model\_data\_10k$accession\_number  
year\_filed <- model\_data\_10k$year\_filed  
model\_data\_10k <- model\_data\_10k %>% select(-c(year\_filed, accession\_number)) # year\_filed is removed so it is not considered by the model  
  
model\_10k <- lm(price\_adjusted\_ratio ~., data = model\_data\_10k, na.action=na.exclude)  
  
save(model\_10k, file = '/Volumes/Buku Gibran/edgar/temp/model\_10k.rda')  
  
load(file = '/Volumes/Buku Gibran/edgar/temp/model\_10k.rda')  
model\_data\_10k$predicted\_ratio <- stats::predict(model\_10k, newdata = model\_data\_10k %>% select(-c(price\_adjusted\_ratio)))  
model\_data\_10k$year\_filed <- year\_filed # add back year\_filed as to help grouping  
model\_data\_10k$accession\_number <- accession\_number  
  
save(model\_data\_10k, file = '/Volumes/Buku Gibran/edgar/temp/model\_data\_10k.rda')  
  
  
local\_df <- head(as.data.frame(summary(model\_10k)$coefficients) %>%  
 tibble::rownames\_to\_column() %>%  
 mutate(absolute\_t\_value = abs(`t value`)) %>%  
 arrange(desc(absolute\_t\_value)) , 1000) %>%  
 mutate(rowname=factor(rowname, levels=rowname)) %>%  
 mutate(significance = case\_when(`Pr(>|t|)` <= 0.001 ~ 'significant\*\*\*', `Pr(>|t|)` <= 0.01 ~ 'significant\*\*', `Pr(>|t|)` <= 0.05 ~ 'significant\*', TRUE ~ 'not significant')) %>%  
 mutate(r\_squared = paste('Multiple R-squared:',as.character(round(summary(model\_10k)$r.squared,3)))) %>%  
 mutate(category = case\_when(grepl("company\_name", rowname, fixed = TRUE) ~ 'Company Feature', grepl("gics\_sector", rowname, fixed = TRUE) ~ 'Sector Feature', TRUE ~ 'Sentiment Feature'))  
  
  
feature\_category <- c('Sentiment Feature', 'Sector Feature', 'Company Feature')  
  
model\_df <- data.frame()  
for(v in 1:length(feature\_category)) {  
 top\_n <- local\_df %>% filter(category == feature\_category[v]) %>% arrange(desc(absolute\_t\_value))  
 top\_n <- top\_n[1:10,]  
 model\_df <- bind\_rows(model\_df, top\_n)  
}  
  
  
ggplot(model\_df, aes(x = reorder(rowname, absolute\_t\_value), y = absolute\_t\_value, fill=significance)) + geom\_bar(stat = "identity") +  
 labs(title ='Model Descriptives: Top 10 features', subtitle = 'groupings on feature type',y = 'Absolute t-value', x = 'Features', caption = paste0('R-squared = ', as.character(round(summary(model\_10k)$r.squared,3)) )) +  
 ylim(0,5) + coord\_flip() +  
 facet\_wrap(~category, nrow = 4, scales = "free")  
  
  
rm(model\_df, local\_df, top\_n)

## Model Evaluation - 10-K

load(file = '/Volumes/Buku Gibran/edgar/temp/model\_data\_10k.rda')  
  
rsq <- function (x, y) cor(x, y) ^ 2 # setup R-squared calculation  
  
# ----- Company Level  
company\_agg <- model\_data\_10k %>%  
 group\_by(year\_filed, company\_name, gics\_sector) %>%  
 summarise(actual\_ratio = mean(price\_adjusted\_ratio),  
 predicted\_ratio = mean(predicted\_ratio))  
  
company\_agg\_rsquared <- round(rsq(company\_agg$actual\_ratio, company\_agg$predicted\_ratio), 3) # calculate R-squared between actual and predicted  
   
company\_agg\_actual <- company\_agg %>% select(-predicted\_ratio) %>% mutate(price\_adjusted\_ratio = actual\_ratio, group = 'Actual') %>% select(-actual\_ratio)  
company\_agg\_predicted <- company\_agg %>% select(-actual\_ratio) %>% mutate(price\_adjusted\_ratio = predicted\_ratio, group = 'Model') %>% select(-predicted\_ratio)  
   
company\_agg <- bind\_rows(company\_agg\_actual, company\_agg\_predicted)  
   
# ----- Plot Actual vs Model  
ggplot(company\_agg, aes(x = year\_filed, y = price\_adjusted\_ratio, color = gics\_sector)) + geom\_line(aes(group = company\_name)) + coord\_cartesian(ylim=c(-50,50)) + facet\_wrap(~group, ncol = 4, scales = "free") + labs(title ='10-K Model Performance: Actual vs Model', subtitle = 'groupings on company Level',y = '% Change in Stock Price', x = 'Year Filed', caption = paste0('R-squared = ', company\_agg\_rsquared))  
  
rm(company\_agg, company\_agg\_rsquared, company\_agg\_actual, company\_agg\_predicted)

# ----- GICS Industry Level  
gics\_agg <- model\_data\_10k %>%  
 group\_by(year\_filed, gics\_sector) %>%  
 summarise(actual\_ratio = mean(price\_adjusted\_ratio),  
 predicted\_ratio = mean(predicted\_ratio))  
  
gics\_agg\_rsquared <- round(rsq(gics\_agg$actual\_ratio, gics\_agg$predicted\_ratio), 3) # calculate R-squared between actual and predicted  
   
gics\_agg\_actual <- gics\_agg %>% select(-predicted\_ratio) %>% mutate(price\_adjusted\_ratio = actual\_ratio, group = 'Actual') %>% select(-actual\_ratio)  
gics\_agg\_predicted <- gics\_agg %>% select(-actual\_ratio) %>% mutate(price\_adjusted\_ratio = predicted\_ratio, group = 'Model') %>% select(-predicted\_ratio)  
   
gics\_agg <- bind\_rows(gics\_agg\_actual, gics\_agg\_predicted)  
   
# ----- Plot Actual vs Model  
ggplot(gics\_agg, aes(x = year\_filed, y = price\_adjusted\_ratio, color = gics\_sector)) + geom\_line(aes(group = gics\_sector)) + coord\_cartesian(ylim=c(-10,10)) + facet\_wrap(~group, ncol = 4, scales = "free") + labs(title ='10-K Model Performance: Actual vs Model', subtitle = 'groupings on GICS Sector Level',y = '% Change in Stock Price', x = 'Year Filed', caption = paste0('R-squared = ', gics\_agg\_rsquared))  
  
rm(gics\_agg, gics\_agg\_rsquared, gics\_agg\_actual, gics\_agg\_predicted)

# ----- Market Level  
market\_agg <- model\_data\_10k %>%  
 group\_by(year\_filed) %>%  
 summarise(actual\_ratio = mean(price\_adjusted\_ratio),  
 predicted\_ratio = mean(predicted\_ratio))  
  
market\_agg\_rsquared <- round(rsq(market\_agg$actual\_ratio, market\_agg$predicted\_ratio), 3) # calculate R-squared between actual and predicted  
   
market\_agg\_actual <- market\_agg %>% select(-predicted\_ratio) %>% mutate(price\_adjusted\_ratio = actual\_ratio, group = 'Actual') %>% select(-actual\_ratio)  
market\_agg\_predicted <- market\_agg %>% select(-actual\_ratio) %>% mutate(price\_adjusted\_ratio = predicted\_ratio, group = 'Model') %>% select(-predicted\_ratio)  
   
market\_agg <- bind\_rows(market\_agg\_actual, market\_agg\_predicted)  
   
# ----- Plot Actual vs Model  
ggplot(market\_agg, aes(x = year\_filed, y = price\_adjusted\_ratio, group =1)) + geom\_line() + coord\_cartesian(ylim=c(-10,10)) + facet\_wrap(~group, ncol = 4, scales = "free") + labs(title ='10-K Model Performance: Actual vs Model', subtitle = 'groupings on Market Level',y = '% Change in Stock Price', x = 'Year Filed', caption = paste0('R-squared = ', market\_agg\_rsquared))  
   
rm(market\_agg, market\_agg\_rsquared, market\_agg\_actual, market\_agg\_predicted)

## Modelling - 10-Q

load(file = '/Volumes/Buku Gibran/edgar/temp/sentiment\_data.rda')  
# ----- Stock Price Modelling  
model\_data\_10q <- sentiment\_data %>%   
 filter(form\_type == "10-Q") %>%  
 select(-c(form\_type, cik, return\_adjusted\_price))  
  
accession\_number <- model\_data\_10q$accession\_number  
year\_filed <- model\_data\_10q$year\_filed  
model\_data\_10q <- model\_data\_10q %>% select(-c(year\_filed, accession\_number)) # year\_filed is removed so it is not considered by the model  
  
model\_10q <- lm(price\_adjusted\_ratio ~., data = model\_data\_10q, na.action=na.exclude)  
  
save(model\_10q, file = '/Volumes/Buku Gibran/edgar/temp/model\_10q.rda')  
  
load(file = '/Volumes/Buku Gibran/edgar/temp/model\_10q.rda')  
model\_data\_10q$predicted\_ratio <- stats::predict(model\_10q, newdata = model\_data\_10q %>% select(-c(price\_adjusted\_ratio)))  
model\_data\_10q$year\_filed <- year\_filed # add back year\_filed as to help grouping  
model\_data\_10q$accession\_number <- accession\_number  
  
save(model\_data\_10q, file = '/Volumes/Buku Gibran/edgar/temp/model\_data\_10q.rda')  
load(file = '/Volumes/Buku Gibran/edgar/temp/model\_data\_10q.rda')  
  
  
summary(model\_10q)  
  
local\_df <- head(as.data.frame(summary(model\_10q)$coefficients) %>%  
 tibble::rownames\_to\_column() %>%  
 mutate(absolute\_t\_value = abs(`t value`)) %>%  
 arrange(desc(absolute\_t\_value)) , 1000) %>%  
 mutate(rowname=factor(rowname, levels=rowname)) %>%  
 mutate(significance = case\_when(`Pr(>|t|)` <= 0.001 ~ 'significant\*\*\*', `Pr(>|t|)` <= 0.01 ~ 'significant\*\*', `Pr(>|t|)` <= 0.05 ~ 'significant\*', TRUE ~ 'not significant')) %>%  
 mutate(r\_squared = paste('Multiple R-squared:',as.character(round(summary(model\_10q)$r.squared,3)))) %>%  
 mutate(category = case\_when(grepl("company\_name", rowname, fixed = TRUE) ~ 'Company Feature', grepl("gics\_sector", rowname, fixed = TRUE) ~ 'Sector Feature', TRUE ~ 'Sentiment Feature'))  
  
  
feature\_category <- c('Sentiment Feature', 'Sector Feature', 'Company Feature')  
  
model\_df <- data.frame()  
for(v in 1:length(feature\_category)) {  
 top\_n <- local\_df %>% filter(category == feature\_category[v]) %>% arrange(desc(absolute\_t\_value))  
 top\_n <- top\_n[1:10,]  
 model\_df <- bind\_rows(model\_df, top\_n)  
}  
  
  
ggplot(model\_df, aes(x = reorder(rowname, absolute\_t\_value), y = absolute\_t\_value, fill=significance)) + geom\_bar(stat = "identity") +  
 labs(title ='10-Q Model Descriptives: Top 10 features', subtitle = 'groupings on feature type',y = 'Absolute t-value', x = 'Features', caption = paste0('R-squared = ', as.character(round(summary(model\_10q)$r.squared,3)) )) +  
 ylim(0,5) + coord\_flip() +  
 facet\_wrap(~category, nrow = 4, scales = "free")  
  
  
rm(model\_df, local\_df, top\_n, sentiment\_data)

## Model Evaluation - 10-Q

# ----- Company Level  
company\_agg <- model\_data\_10q %>%  
 group\_by(year\_filed, company\_name, gics\_sector) %>%  
 summarise(actual\_ratio = mean(price\_adjusted\_ratio),  
 predicted\_ratio = mean(predicted\_ratio))  
  
company\_agg\_rsquared <- round(rsq(company\_agg$actual\_ratio, company\_agg$predicted\_ratio), 3) # calculate R-squared between actual and predicted  
   
company\_agg\_actual <- company\_agg %>% select(-predicted\_ratio) %>% mutate(price\_adjusted\_ratio = actual\_ratio, group = 'Actual') %>% select(-actual\_ratio)  
company\_agg\_predicted <- company\_agg %>% select(-actual\_ratio) %>% mutate(price\_adjusted\_ratio = predicted\_ratio, group = 'Model') %>% select(-predicted\_ratio)  
   
company\_agg <- bind\_rows(company\_agg\_actual, company\_agg\_predicted)  
   
# ----- Plot Actual vs Model  
ggplot(company\_agg, aes(x = year\_filed, y = price\_adjusted\_ratio, color = gics\_sector)) + geom\_line(aes(group = company\_name)) + coord\_cartesian(ylim=c(-50,50)) + facet\_wrap(~group, ncol = 4, scales = "free") + labs(title ='10-Q Model Performance: Actual vs Model', subtitle = 'groupings on company Level',y = '% Change in Stock Price', x = 'Year Filed', caption = paste0('R-squared = ', company\_agg\_rsquared))  
  
rm(company\_agg, company\_agg\_rsquared, company\_agg\_actual, company\_agg\_predicted)

# ----- GICS Industry Level  
gics\_agg <- model\_data\_10q %>%  
 group\_by(year\_filed, gics\_sector) %>%  
 summarise(actual\_ratio = mean(price\_adjusted\_ratio),  
 predicted\_ratio = mean(predicted\_ratio))  
  
gics\_agg\_rsquared <- round(rsq(gics\_agg$actual\_ratio, gics\_agg$predicted\_ratio), 3) # calculate R-squared between actual and predicted  
   
gics\_agg\_actual <- gics\_agg %>% select(-predicted\_ratio) %>% mutate(price\_adjusted\_ratio = actual\_ratio, group = 'Actual') %>% select(-actual\_ratio)  
gics\_agg\_predicted <- gics\_agg %>% select(-actual\_ratio) %>% mutate(price\_adjusted\_ratio = predicted\_ratio, group = 'Model') %>% select(-predicted\_ratio)  
   
gics\_agg <- bind\_rows(gics\_agg\_actual, gics\_agg\_predicted)  
   
# ----- Plot Actual vs Model  
ggplot(gics\_agg, aes(x = year\_filed, y = price\_adjusted\_ratio, color = gics\_sector)) + geom\_line(aes(group = gics\_sector)) + coord\_cartesian(ylim=c(-10,10)) + facet\_wrap(~group, ncol = 4, scales = "free") + labs(title ='10-Q Model Performance: Actual vs Model', subtitle = 'groupings on GICS Sector Level',y = '% Change in Stock Price', x = 'Year Filed', caption = paste0('R-squared = ', gics\_agg\_rsquared))  
  
rm(gics\_agg, gics\_agg\_rsquared, gics\_agg\_actual, gics\_agg\_predicted)

# ----- Market Level  
market\_agg <- model\_data\_10q %>%  
 group\_by(year\_filed) %>%  
 summarise(actual\_ratio = mean(price\_adjusted\_ratio),  
 predicted\_ratio = mean(predicted\_ratio))  
  
market\_agg\_rsquared <- round(rsq(market\_agg$actual\_ratio, market\_agg$predicted\_ratio), 3) # calculate R-squared between actual and predicted  
   
market\_agg\_actual <- market\_agg %>% select(-predicted\_ratio) %>% mutate(price\_adjusted\_ratio = actual\_ratio, group = 'Actual') %>% select(-actual\_ratio)  
market\_agg\_predicted <- market\_agg %>% select(-actual\_ratio) %>% mutate(price\_adjusted\_ratio = predicted\_ratio, group = 'Model') %>% select(-predicted\_ratio)  
   
market\_agg <- bind\_rows(market\_agg\_actual, market\_agg\_predicted)  
   
# ----- Plot Actual vs Model  
ggplot(market\_agg, aes(x = year\_filed, y = price\_adjusted\_ratio, group =1)) + geom\_line() + coord\_cartesian(ylim=c(-10,10)) + facet\_wrap(~group, ncol = 4, scales = "free") + labs(title ='10-Q Model Performance: Actual vs Model', subtitle = 'groupings on Market Level',y = '% Change in Stock Price', x = 'Year Filed', caption = paste0('R-squared = ', market\_agg\_rsquared))  
   
rm(market\_agg, market\_agg\_rsquared, market\_agg\_actual, market\_agg\_predicted)