News Headlines as predictors of Flex company stock movement: Using Unigrams, Bigrams, Trigrams, and Fourgrams

Raghu Kowshik

January 25, 2017

INTRODUCTION

Stock market trends are volatile. Financial analysts, researchers, and financial executives of firms are interested in knowing how to predict stock price movement. In recent years, machine learning algorithms have been widely used to come up with prediction approaches. This project attempts to use non-quantifiable data such as the daily news headlines to predict a firm's future stock trend. Based on the assumption that news headlines impact the stock market trends, this project attempts to study the effectiveness of a machine learning algorithm approach in predicting a firm's stock movement. To show this, this researcher created five generalized linear models with cross validation and ridge regression. The validity and accuracy of the models is tested using ROC and AUC scores. The text processing algorithm processes the headlines into a document term matrix. The stock price trend is coded as 0 and 1. Dual density plots were generated to examine whether the headlines affect the stock price flag (1 = Up, 0 = down). Results show a clear separateion in the dual density plots, however, the best accuracy rate was 56.05% and the lowest accuracy was 51.51% as indicated by the AUC score of the five models.

DATA COLLECTION

The stock price data for Flex Ltd company was collected from http://www.nasdaq.com/symbol/flex/historical for the past 13 months, from December 1, 2015 to December 30, 2016. The news headlines were collected for the same period from http://www.newsmax.com/Archives/Newsfront/16/2016/12/. The daily stock price data contains five attributes namely Open, High, Low, Close, and Volume, Adjusted Close. Only the open and close price values were used in this project to code the stock movement as going up or down. The two data chunks were merged to form a single data file in csv format. The dataset contains 175 observations. The newsmax.com headlines consisted of 20 headlines per day. After merging the two data sets, the data set used for this projECT had the following variables: date, close, volume, open, high, ha1, hl2, hl3, hl4, hl5, hl6, hl7, hl8, hl9, hl10, hl11, hl12, hl13, hl14, hl15, hl16, hl17, hl18, hl19, hl20. The h1 through h20 and headline text each day.

FEATURE ENGINEERING

Before importing the dataset into RStudio, a feature, a new variable called "label", was added. The "label" variable was assigned binary values 0 or 1. If the stock value at close was less than the open value, then the variable "label" was assigned 0, if the stock value at close was greater than or equal to the open value, then the variable "label" was assigned a 1. The final data set "headlines" has the following variables: date, close, volume, open, high, label, ha1, hl2, hl3, hl4, hl5, hl6, hl7, hl8, hl9, hl10, hl11, hl12, hl13, hl14, hl15, hl16, hl17, hl18, hl19, hl20.

METHODOLOGY

This project attempts to predict Flex (NASDAQ: FLEX) company stock movement using a variety of R tools including RWeka, tm, and glmnet packages. The following packages were loaded using the install.packages utility and the library command in RStudio.

NLP, tm, RWeka, magrittr, Matrix, glmnet, ROCR, ggplot2, pROC.

The tm package was used for text processing and creating a document-term matrix (dtm) of news headlines. The glmnet package was used to perform logistic regression modelling. The ROCR package was used to generate ROC curves and assess performance using the area under curve (AUC). The RWeka package was used to perform a variety of text processing tasks including NGramTokenizing.

The glmnet is a linear regression library for R and is central to this study. It creates regression models, and predictions from those models. The regression model exploits the sparsity in the input data. Regularization is a technique used in linear regression with p features and n observations. To minimize the sum of squared errors and to reduce overfitting, the regularization method adds a penalty term. This is called ridge regression which shrinks the correlated variables toward each other. Lambda is the shrinkage factor used in determining the text feature predictive of the stock trend. Rather than specifying a fixed lambda value for shrinkage, cross-validation is used to automtically select the best lambda value.

LOAD CSV DATA INTO R DATA FRAME

```
headlines <- read.csv(file="headlines.csv", stringsAsFactors = FALSE)
```

DATA WRANGLING (PRE-PROCESSING)

The following data cleanup steps were performed.

1: Make "Date" column a date object

```
headlines$date <- as.Date(headlines$date)
```

2: Remove unused variables

```
headlines$volume = NULL
headlines$high = NULL
headlines$low = NULL
headlines$close = NULL
headlines$open = NULL
```

3: Combine headlines into one text group for each day and add sentence separation

```
headlines$all <- paste(headlines$hl1, headlines$hl2, headlines$hl3, headlines$hl4,
                       headlines$hl5, headlines$hl6, headlines$hl7, headlines$HL8,
                       headlines$hl9, headlines$hl10, headlines$hl11, headlines$hl12,
                       headlines$hl13, headlines$hl14, headlines$hl15, headlines$hl16,
                       headlines$hl17, headlines$hl18, headlines$hl19, headlines$hl20,
                       sep=" <s> ")
```

4: Remove all punctuation except headline separators, add a new variable, "all"

```
headlines$all <- gsub("([<>])|[[:punct:]]", "\\1", headlines$all)
```

5: Reduce to olny three columns we need

```
headlines <- headlines[, c('date', 'label', 'all')]
```

6: Eliminate unnecessary words and space, transform to lower

Via a corpus object, using the "tm" package, text headlines are converted into a document-term matrix. Each row of the document-term matrix is the combined headlines for each day. Columns will be frequency counts of unigrams. In the tm package, before converting

the headline text into a term matrix, the control object (see R code below) will tell the DocumentTermMatrix() function to remove punctuation and numbers, convert the text to lowercase, and remove common English words.

```
control_c <- list(stopwords=c(stopwords(kind = 'SMART'), '<s>'))
                  # exclude stopwords and headline tokens
dtm <- Corpus(VectorSource(headlines$all)) %>%
  tm_map(removeNumbers) %>%
  tm_map(stripWhitespace) %>%
  tm_map(content_transformer(tolower)) %>%
  DocumentTermMatrix(control=control_c)
```

DATA SET PREPARATION

Using a date as an index marker, divide the dataset into two.

```
split_index_1 <- headlines$date <= '2016-09-30'
```

Select the index so that approximately 70% of the data can be used for training and 30% for testing.

```
ytrain <- as.factor(headlines$label[split_index_1])
xtrain <- Matrix(as.matrix(dtm)[split_index_1, ], sparse=TRUE)

ytest <- as.factor(headlines$label[!split_index_1])
xtest <- Matrix(as.matrix(dtm)[!split_index_1, ], sparse=TRUE)
```

MODEL BUILDING AND TESTING

Five linear regression models were tested. The following sections illustrate the R code and the results. The glmnet model, using ridge rigression, was trained using the train data set. The test data set was used for generating the predictions. Cross validation was selected for the glmnet model, so that the R program selected the best lambda value for model fit.

MODEL 1

1. Fit glmnet model, train the model using train data set

In this model, alpha = 0 indicates that this is a ridge regression model.

```
glmnet.fit.1 <- cv.glmnet(x=xtrain, y=ytrain, family='binomial', alpha=0)
```

2. Generate predictions using the test data set

As stated previously, Lambda is the shrinkage factor used in determining the text feature predictive of the stock trend. Rather than specifying a fixed lambda value for shrinkage, cross-validation is used to automtically select the best lambda value.

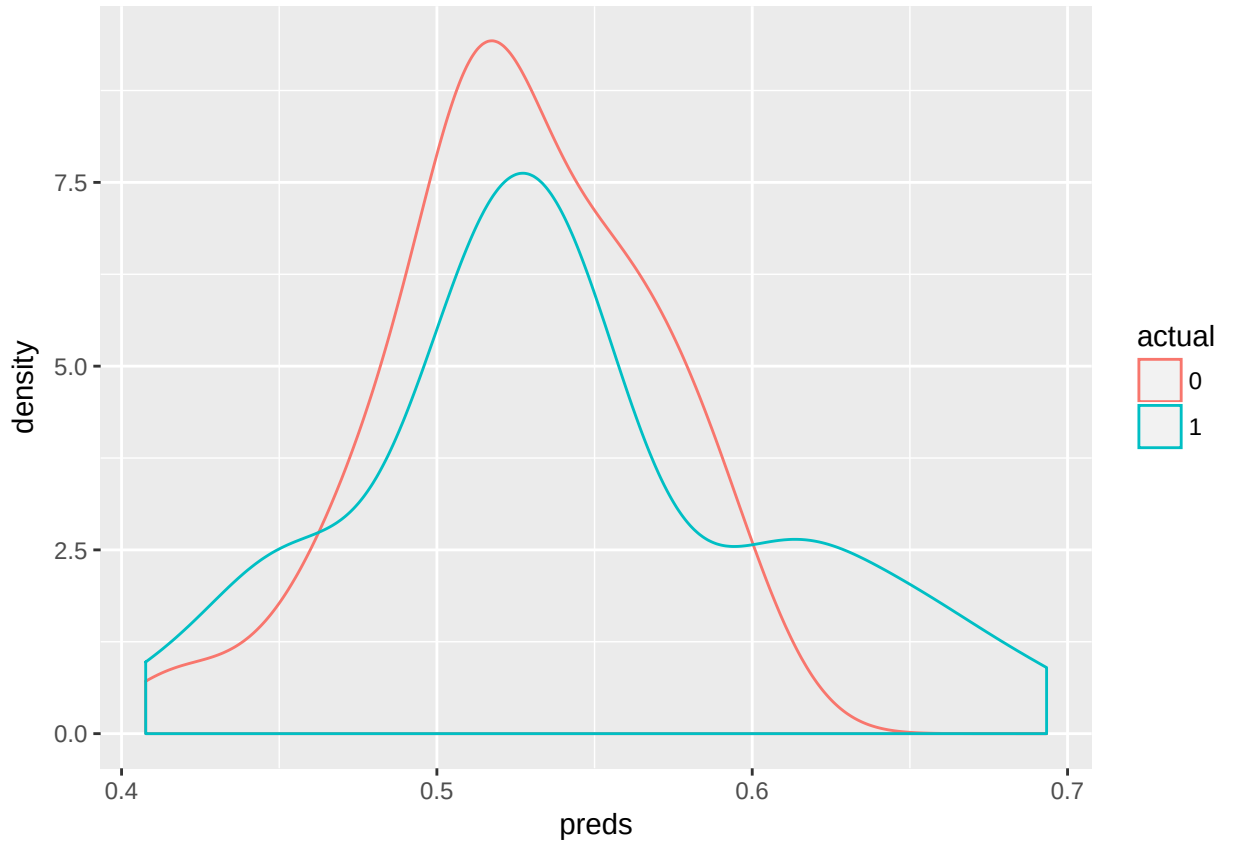
```
preds <- predict(glmnet.fit.1, newx=xtest, type='response', s='lambda.min')
```

3. Put results into dataframe for plotting

```
results1 <- data.frame(pred=preds, actual=ytest)
```

4. Generate dual density plot

Dual density plot of the predicted probabilites was generated, for each of the true values 0 (stock is down) or 1 (stock is up or unchanged). There is discernable separation between the 0 plot and the 1 plot. If these two plots were identical, it would amount to a 50%-50% probability of guessing the stock trend going up or down. Since the two plots are not identical, we have a reasonable prediction model.



5. Model 1 Performance Assessment

The dual density plot shows that we have a reasonable prediction model. How reasonable? How accurate is the prediction? The ROCR package was used to assess model performance. Two performance objects were created: one for the true positive/false positive rates, and the second performance object for the AUC score.

a). AUC Score

AUC, or Area Under Curve, is a metric for binary classification. It's the second most popular one, after accuracy.

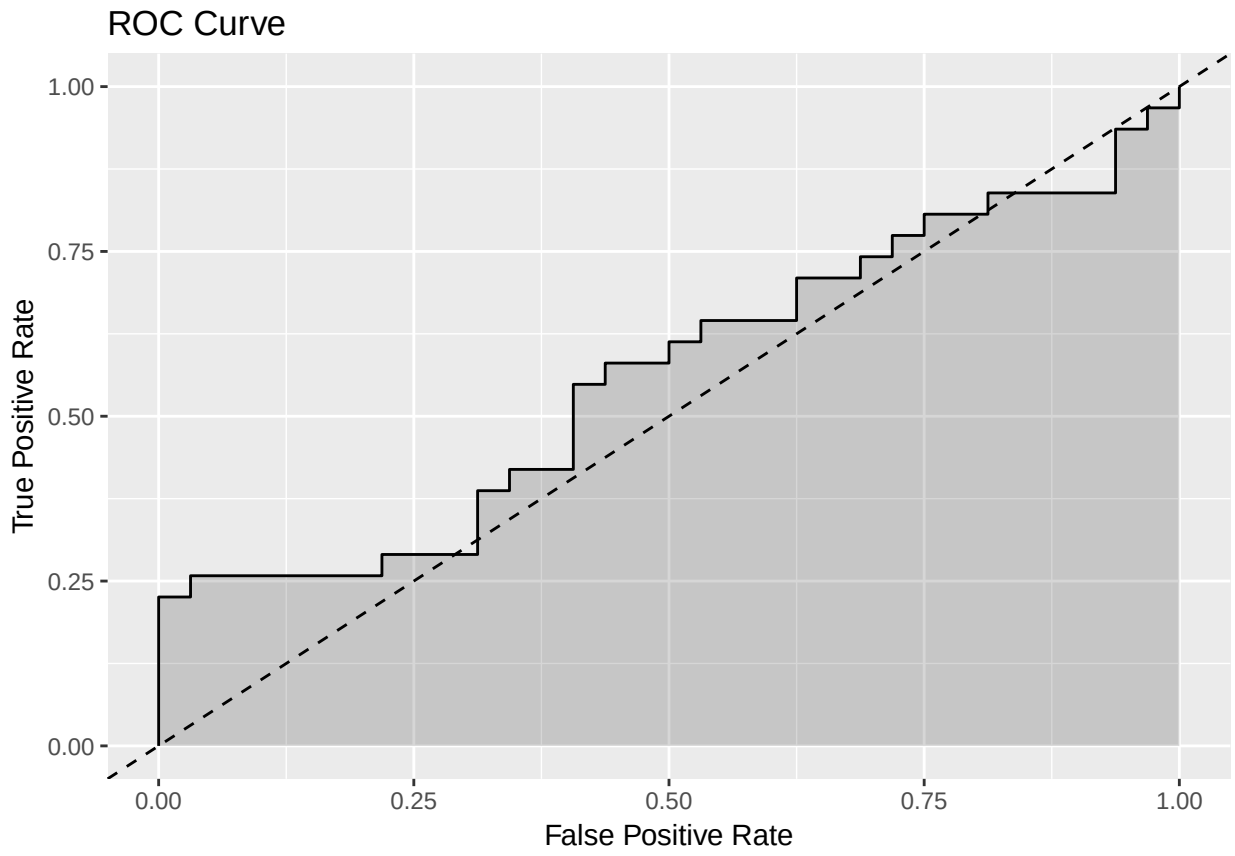
Accuracy deals with ones and zeros, meaning either the class label is right or it is not. But many classifiers are able to quantify their uncertainty about the answer by outputting a probability value. To compute accuracy from probabilities a threshold is needed to decide when zero turns into one. The most natural threshold is 0.5.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5605  0.5605  0.5605  0.5605  0.5605  0.5605
```

With an AUC score of 0.5565, we have a 55.65% probability that the true positive rate is accurate. This is not great, but better than random guessing.

b). False Positive/True Positive Rate

AUC considers all possible thresholds. Various thresholds result in different true positive/false positive rates. As the threshold is decreased, the result is more true positives, but also more false positives.The relation between them for Model 1 can be plotted as shown by the ROC Curve below. The dotted line is the threshold curve. As we can see, prediction curve is not coinciding with the threshold. We have a not so great, but reasonable prediction model, an indicator of future Flex stock trend. Of course, one would not be advised to make trading decisions on the basis of this prediction!



MODEL 2

Model 2 also uses individul words, just like model 1, but takes the tokenizer approach to delienate the individual words.

```
unigramTokenizer <- function(x) NGramTokenizer(x, Weka_control(min = 1, max = 1))
```

1. Give a boundary for the tokenizer.

We will only use unigrams that appear in at least 20 of the documents. We won't use unigrams that appear in more than 120 of the documents.

```
control <- list(tokenize=unigramTokenizer, bounds = list(global = c(20, 120)))
```

2. Convert the headlines into document term matrix. Remove numbers, strip whitespace, transform to lower case.

```
dtm <- Corpus(VectorSource(headlines$all)) %>%
  tm_map(removeNumbers) %>%
```

```
  tm_map(stripWhitespace) %>%
  tm_map(content_transformer(tolower)) %>%
  DocumentTermMatrix(control=control)
```

3. Using a date as an index marker, divide the dataset into two.

```
split_index_2 <- headlines$date <= '2016-09-30'

ytrain <- as.factor(headlines$label[split_index_2])
xtrain <- Matrix(as.matrix(dtm)[split_index_2, ], sparse=TRUE)

ytest <- as.factor(headlines$label[!split_index_2])
xtest <- Matrix(as.matrix(dtm)[!split_index_2, ], sparse=TRUE)
```

4. Fit glmnet model, train the model using train data set

```
glmnet.fit.2 <- cv.glmnet(x=xtrain, y=ytrain, family='binomial', alpha=0)
```

5. Generate predictions using the test data set

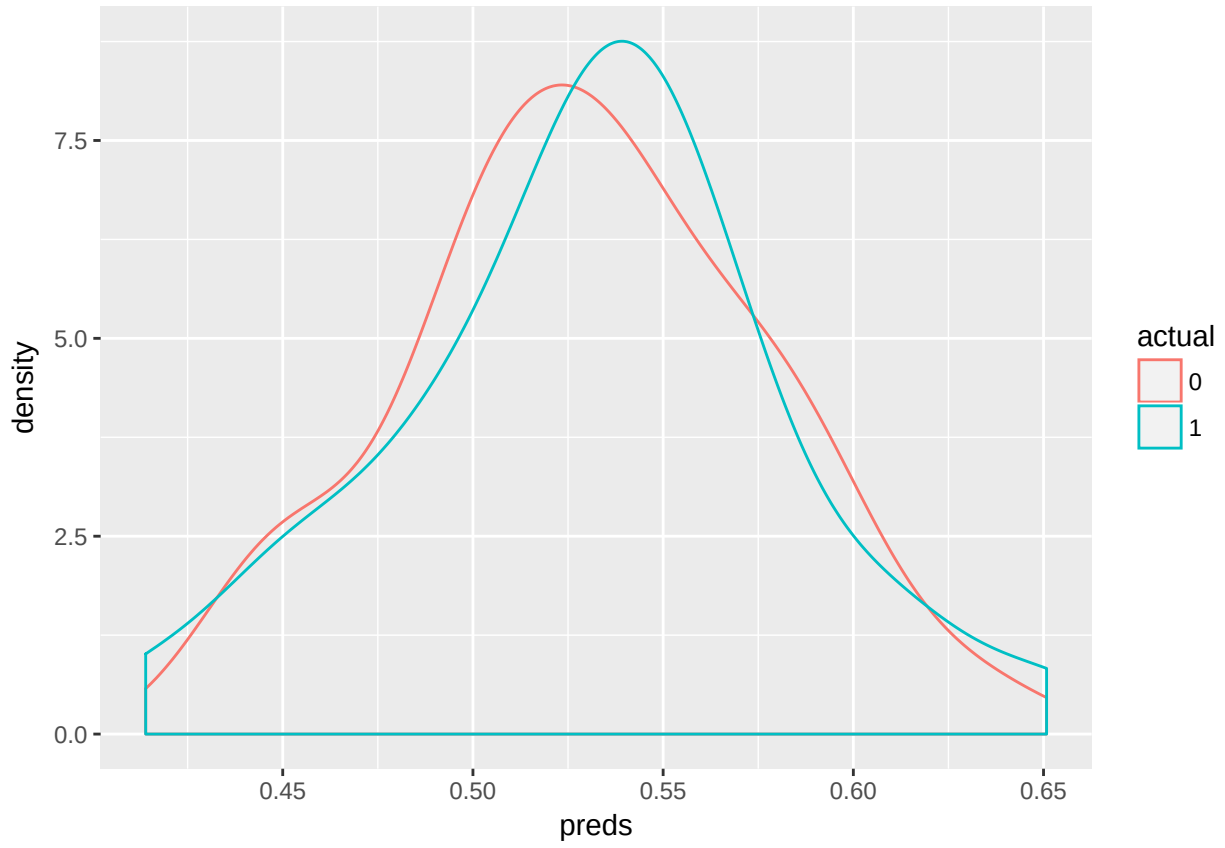
```
preds <- predict(glmnet.fit.2, newx=xtest, type='response', s="lambda.min")
```

6. Put results into dataframe for plotting

```
results2 <- data.frame(pred=preds, actual=ytest)
```

7. Generate dual density plot.

Dual density plot of the predicted probabilites for Model 2 was generated, for each of the true values 0 (stock is down) or 1 (stock is up or unchanged). Again, in this model also, there is discernable separation between the 0 plot and the 1 plot. As stated previously, if these two plots were identical, it would amount to a 50%-50% probability of guessing the stock trend going up or down. Since the two plots are not identical, we have a reasonable prediction model.



8. Model 2 Performance Assessment

As before, the ROCR package was used to assess model performance. Two performance objects were created: one for the true positive/false positive rates, and the second performance object for the AUC score.

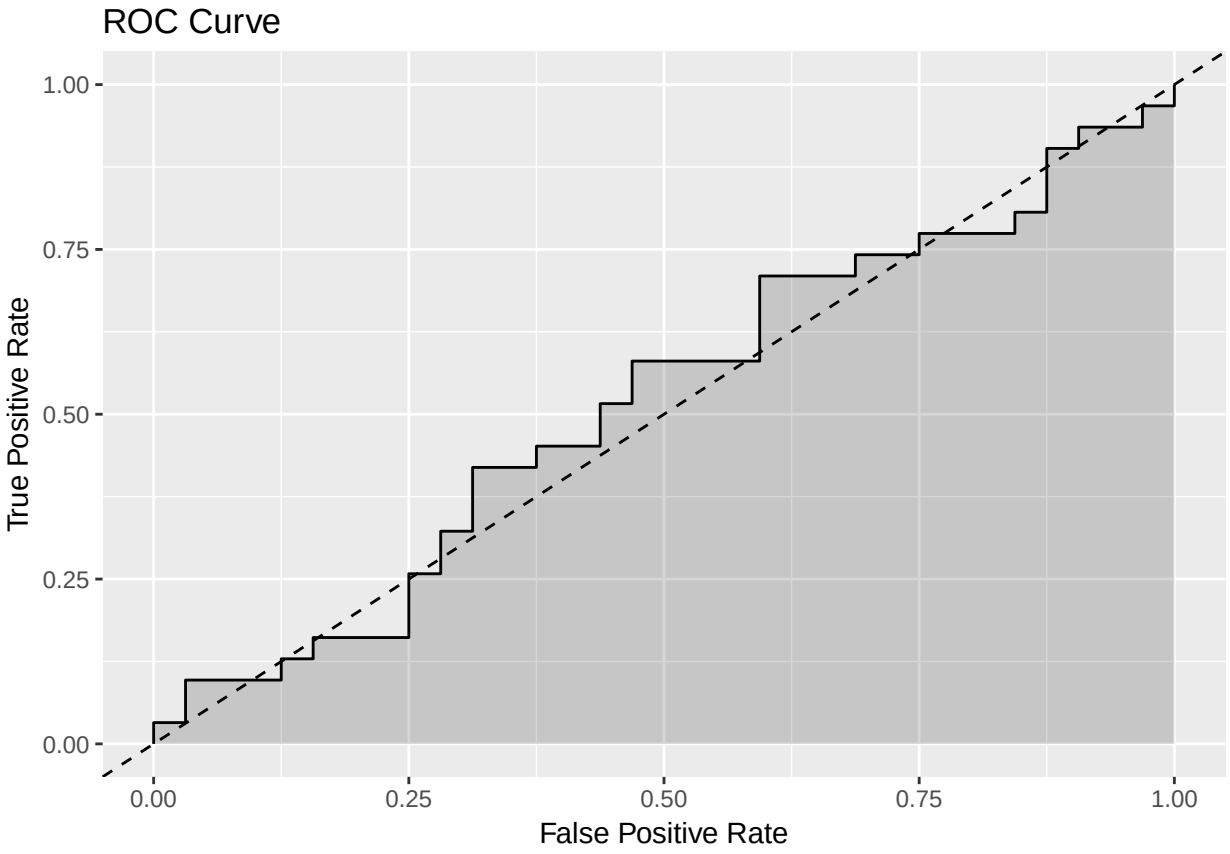
a). AUC Score

With an AUC score of 0.5121, we have a 51.21% probability that the true positive rate is accurate. This is not great, but marginally better than random guessing.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5181  0.5181  0.5181  0.5181  0.5181  0.5181
```

b). False Positive/True Positive Rate

The relation between False Positive/Tru Positive rate for Model 2 is shown by the ROC Curve below. The dotted line is the threshold curve. As we can see, prediction curve is not coinciding with the threshold. We have a marginal but not so great, prediction model, an indicator of future Flex stock trend. Of course, one would not be advised to make trading decisions on the basis of this prediction!



MODEL 3

Model 3 uses bigrams. Uses tokenizer to create bigrams.

```
bigramTokenizer <- function(x) NGramTokenizer(x, Weka_control(min = 2, max = 2))
```

1. Give a boundary for the tokenizer.

We will only use bigrams that appear in at least 20 of the documents. We won't use bigrams that appear in more than 100 of the documents.

```
control <- list(tokenize=unigramTokenizer, bounds = list(global = c(20, 100)))
```

2. Convert the headlines into document term matrix. Remove numbers, strip whitespace, transform to lower case.

```
dtm <- Corpus(VectorSource(headlines$all)) %>%
  tm_map(removeNumbers) %>%
  tm_map(stripWhitespace) %>%
```

```
  tm_map(content_transformer(tolower)) %>%
  DocumentTermMatrix(control=control)
```

3. Using a date as an index marker, divide the dataset into two.

```
split_index_3 <- headlines$date <= '2016-09-30'

ytrain <- as.factor(headlines$label[split_index_3])
xtrain <- Matrix(as.matrix(dtm)[split_index_3, ], sparse=TRUE)

ytest <- as.factor(headlines$label[!split_index_3])
xtest <- Matrix(as.matrix(dtm)[!split_index_3, ], sparse=TRUE)
```

4. Fit the glmnet model, Train the model using the train dataset

```
glmnet.fit.3 <- cv.glmnet(x=xtrain, y=ytrain, family='binomial', alpha=0)
```

5. Generate predictions using the test data set

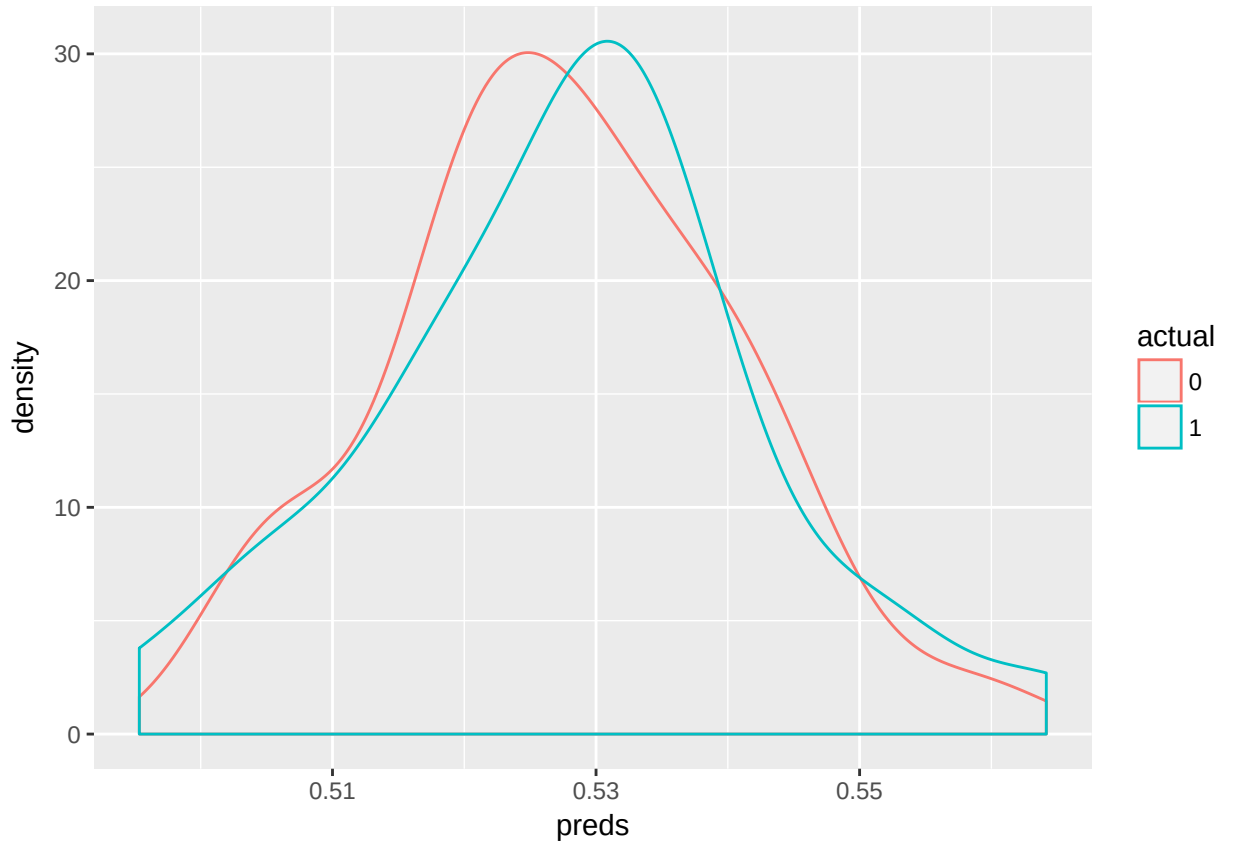
```
preds <- predict(glmnet.fit.3, newx=xtest, type='response', s="lambda.min")
```

6. Put results into dataframe for plotting

```
results3 <- data.frame(pred=preds, actual=ytest)
```

7. Generate dual density plot.

For each of the true values 0 (stock is down) or 1 (stock is up or unchanged), the dual density plot of the predicted probabilites for Model 3 is shown below. The plot shows discernable separation between the 0 plot and the 1 plot. As stated previously, Model 3 is also better than the 50%-50% probability of guessing the stock trend going up or down. We have a reasonable prediction model. How reasonable? The performance assessment below shows the accuracy of Model 3.



###8. Model 3 Performance Assessment

As before, the ROCR package was used to assess model performance. Two performance objects were created: one for the true positive/false positive rates, and the second performance object for the AUC score.

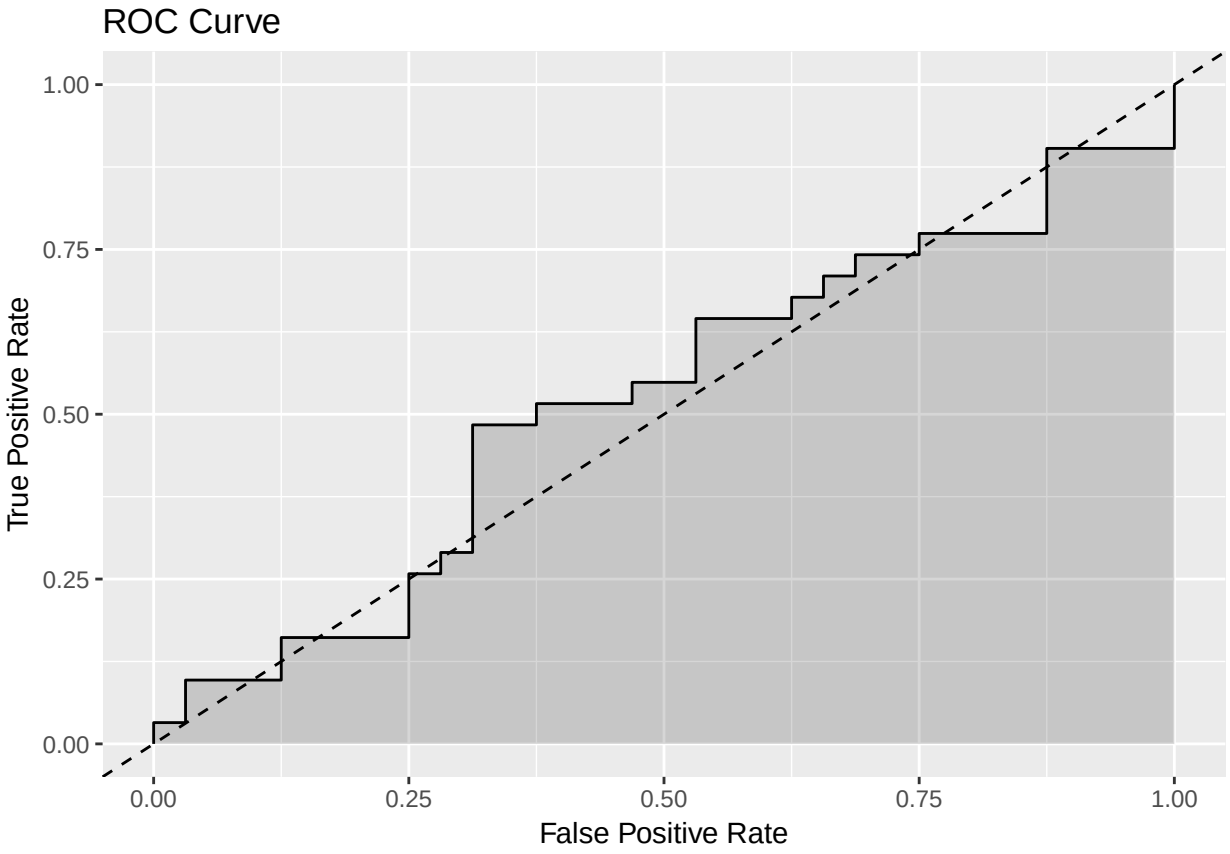
a). AUC Score

With an AUC score of 0.5192, we have a 51.92% probability that the true positive rate for Model 3 is accurate. Model 3 is marginally better than Model 2, and also better than random guessing.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5202  0.5202  0.5202  0.5202  0.5202  0.5202
```

b). False Positive/True Positive Rate

The relation between False Positive/Tru Positive rate for Model 3 is shown by the ROC Curve below. The dotted line is the threshold curve. As we can see, prediction curve is not coinciding with the threshold. We have a marginal, but not so great, prediction model, an indicator of future Flex stock trend. Of course, one would not be advised to make trading decisions on the basis of this prediction!



MODEL 4

Model 4 uses trigrams. Uses tokenizer to create trigrams.

```
bigramTokenizer <- function(x) NGramTokenizer(x, Weka_control(min = 3, max = 3))
```

1. Give a boundary for the tokenizer.

We will only use trigrams that appear in at least 20 of the documents. We won't use trigrams that appear in more than 100 of the documents.

```
control <- list(tokenize=unigramTokenizer, bounds = list(global = c(20, 100)))
```

2. Convert the headlines into document term matrix. Remove numbers, strip whitespace, transform to lower case.

```
dtm <- Corpus(VectorSource(headlines$all)) %>%
  tm_map(removeNumbers) %>%
  tm_map(stripWhitespace) %>%
```

```
  tm_map(content_transformer(tolower)) %>%
  DocumentTermMatrix(control=control)
```

3. Using a date as an index marker, divide the dataset into two.

```
split_index_4 <- headlines$date <= '2016-09-30'

ytrain <- as.factor(headlines$label[split_index_4])
xtrain <- Matrix(as.matrix(dtm)[split_index_4, ], sparse=TRUE)

ytest <- as.factor(headlines$label[!split_index_4])
xtest <- Matrix(as.matrix(dtm)[!split_index_4, ], sparse=TRUE)
```

4. Fit the glmnet model, Train the model using train dataset

```
glmnet.fit.4 <- cv.glmnet(x=xtrain, y=ytrain, family='binomial', alpha=0)
```

5. Generate predictions using the test data set

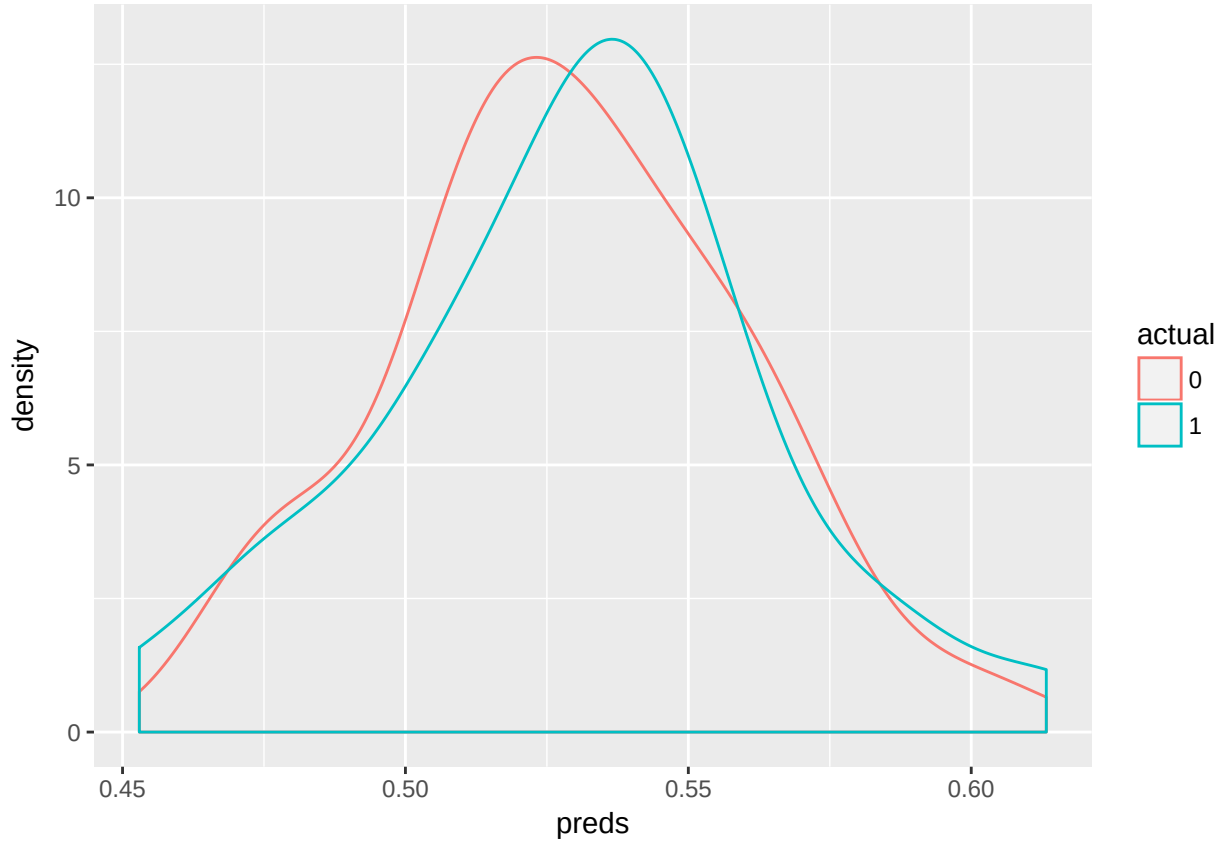
```
preds <- predict(glmnet.fit.4, newx=xtest, type='response', s="lambda.min")
```

6. Put results into dataframe for plotting

```
results4 <- data.frame(pred=preds, actual=ytest)
```

7. Generate dual density plot.

Figure below shows the density plot of the predicted probabilites for Model 4. As stated previously, This is the plot for each of the true values 0 (stock is down) or 1 (stock is up or unchanged). There is discernable separation between the 0 plot and the 1 plot. As observed previously, if these two plots were identical, it would amount to a 50%-50% probability of guessing the stock trend going up or down. We have a reasonable prediction model. Performance assessment below shows the accuracy of the model.



###8. Model 4 Performance Assessment

Again, the ROCR package was used to assess model performance. Two performance objects were created: one for the true positive/false positive rates, and the second performance object for the AUC score.

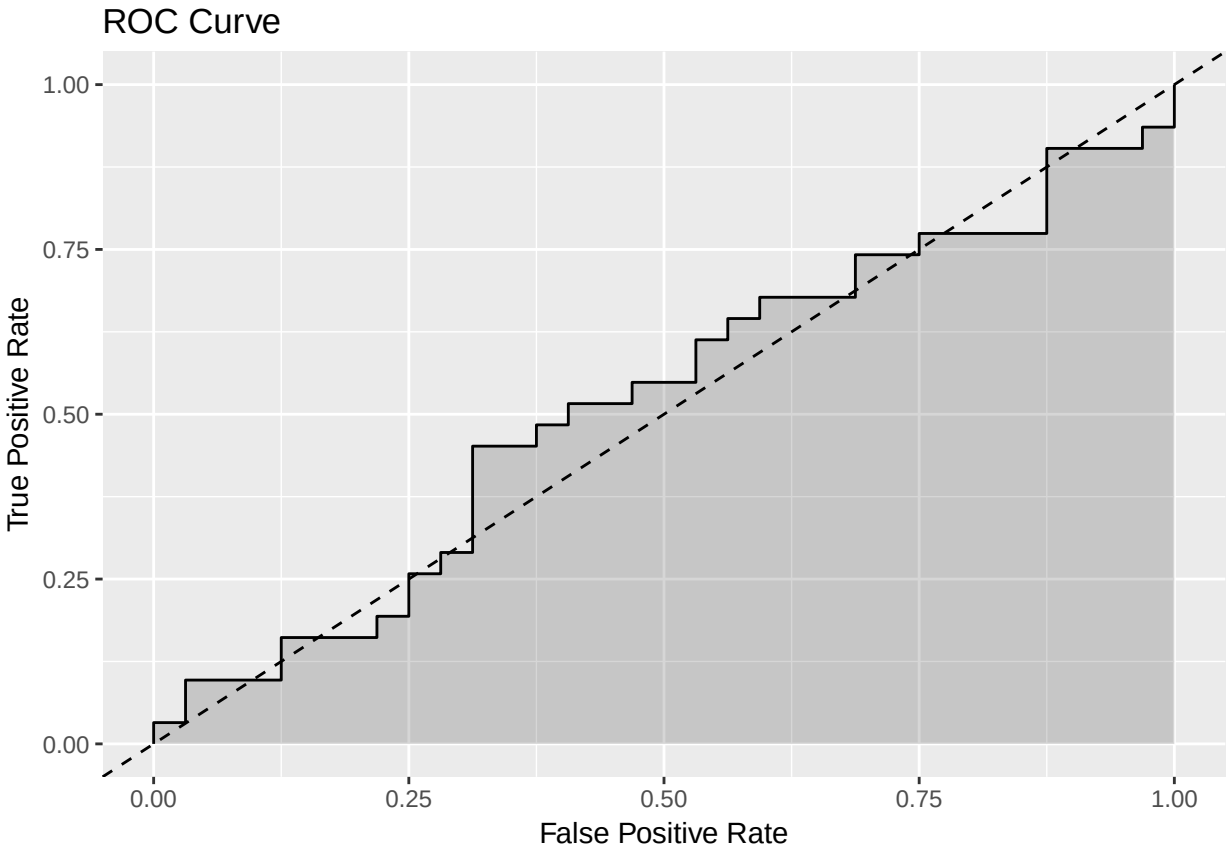
a). AUC Score

With an AUC score of 0.5192, we have a 51.92% probability that the true positive rate for Model 4 is accurate. This is the same performance as Model 3 and also better than random guessing.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5181  0.5181  0.5181  0.5181  0.5181  0.5181
```

b). False Positive/True Positive Rate

The relation between False Positive/Tru Positive rate for Model 4 is shown by the ROC Curve below. The prediction curve is not coinciding with the threshold. We have a marginally better than random guess, but not so great, prediction model, an indicator of future Flex stock trend. Of course, one would not be advised to make trading decisions on the basis of this prediction!



MODEL 5

Model 5 uses fourgrams. Uses tokenizer to create fourgrams.

```
bigramTokenizer <- function(x) NGramTokenizer(x, Weka_control(min = 4, max = 4))
```

1. Give a boundary for the tokenizer.

We will only use fourgrams that appear in at least 20 of the documents. We won't use fourgrams that appear in more than 120 of the documents.

```
control <- list(tokenize=unigramTokenizer, bounds = list(global = c(20, 120)))
```

2. Convert the headlines into document term matrix. Remove numbers, strip whitespace, transform to lower case.

```
dtm <- Corpus(VectorSource(headlines$all)) %>%
  tm_map(removeNumbers) %>%
  tm_map(stripWhitespace) %>%
```

```
  tm_map(content_transformer(tolower)) %>%
  DocumentTermMatrix(control=control)
```

3. Using a date as an index marker, divide the dataset into two.

```
split_index_5 <- headlines$date <= '2016-09-30'

ytrain <- as.factor(headlines$label[split_index_5])
xtrain <- Matrix(as.matrix(dtm)[split_index_5, ], sparse=TRUE)

ytest <- as.factor(headlines$label[!split_index_5])
xtest <- Matrix(as.matrix(dtm)[!split_index_5, ], sparse=TRUE)
```

4. Fit the glmnet model, Train the model using train dataset

```
glmnet.fit.5 <- cv.glmnet(x=xtrain, y=ytrain, family='binomial', alpha=0)
```

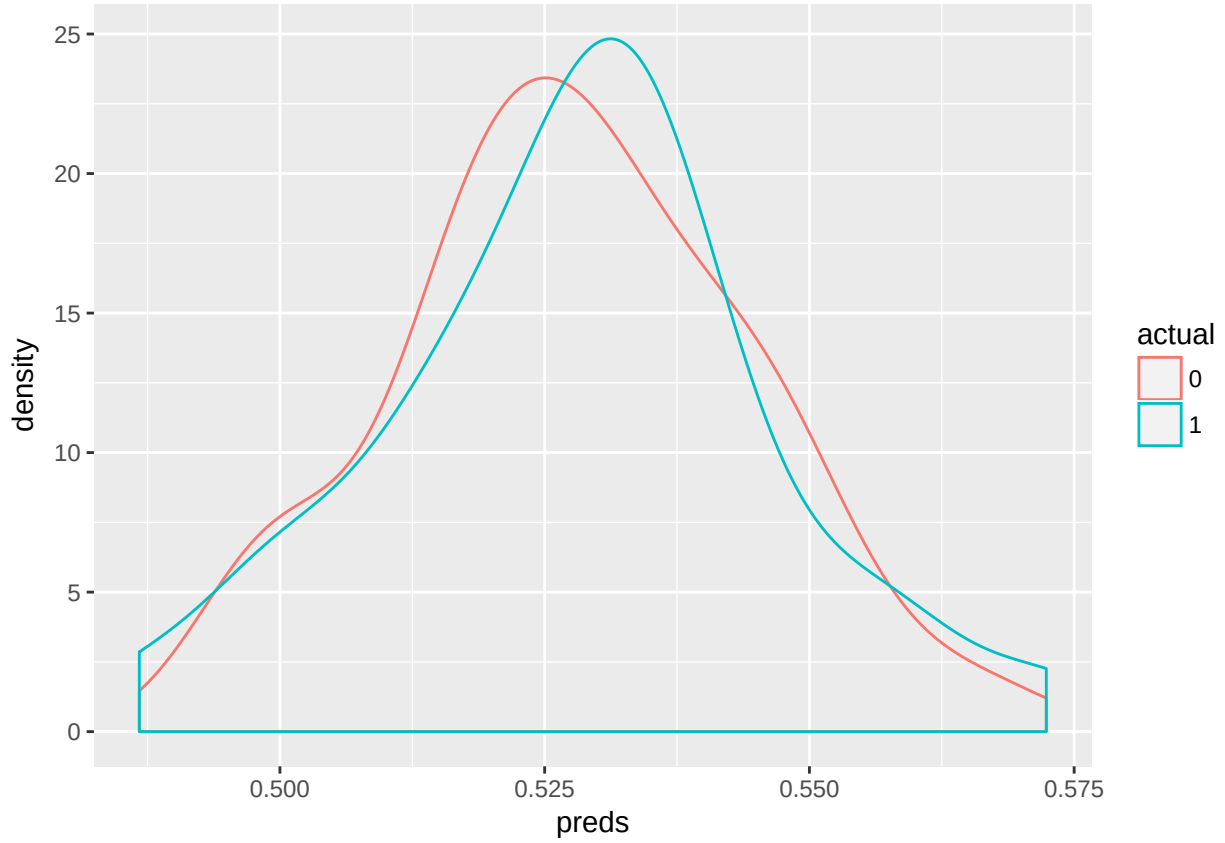
5. Generate predictions using the test data set

```
preds <- predict(glmnet.fit.5, newx=xtest, type='response', s="lambda.min")
```

6. Put results into dataframe for plotting

```
results5 <- data.frame(pred=preds, actual=ytest)
```

7. Generate dual density plot.



Model 5 Performance Assessment

Using the ROCR package, two performance objects were created: one for the true positive/false positive rates, and the second performance object for the AUC score.

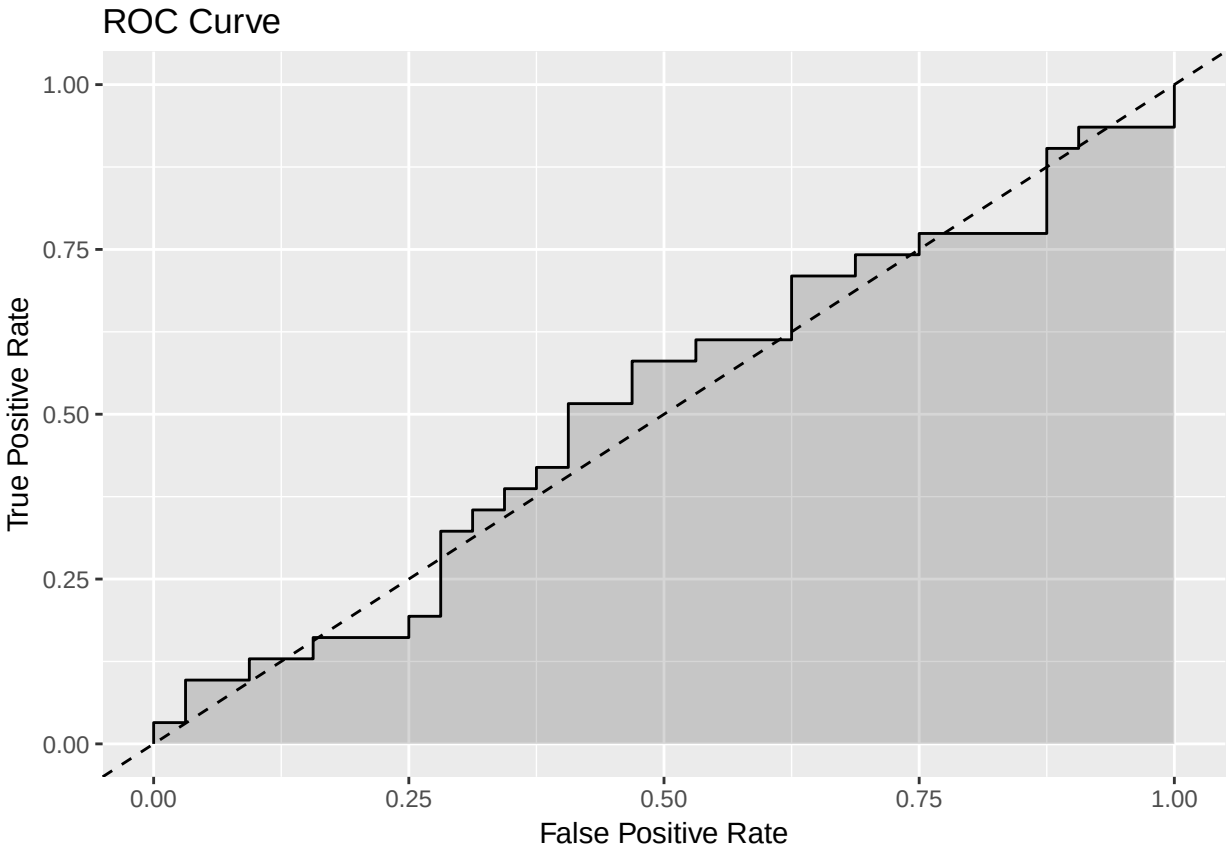
a). AUC Score

With an AUC score of 0.5121, we have a 51.21% probability that the true positive rate for Model 4 is accurate. This is marginally less than Model 4, but better than random guessing.

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5121  0.5121  0.5121  0.5121  0.5121  0.5121
```

b). False Positive/True Positive Rate

The relation between False Positive/Tru Positive rate for Model 5 is shown by the ROC Curve below. The prediction curve is not coinciding with the threshold. We have a marginally better than random guess, but not so great, prediction model, an indicator of future Flex stock trend. Of course, one would not be advised to make trading decisions on the basis of this prediction!



LIMITATIONS OF THIS STUDY

This study has few limitations. The dataset is small, only 274 observations, 3 variables used in the analysis, and 20+ unused variables. Although 5 models were tested in this study, the methodology was identical. The predictions of stock movement up or down, indicated by 1 or 0, were separated, indicating that there is a significant difference in the predictions for stock price going up and stock price going down.

Model AUC score

Model 1: 0.56

Model 2: 0.51

Model 3: 0.52

Model 4: 0.52

Model 5: 0.51

Given that the AUC score ranging from 0.51 to 0.56 is slightly above the thershold value of 0.5, it is reasonable to conclude that the prediction is better than random guessing. However, since we do not know the standard deviation of this score, One might argue that the separation was achieved by chance.

The training data set had 219 observations, and the test data set 55 observations. Considering that this is text processing, perhaps a larger data set of 5 years duration might have better variation in text and offer better, more accurate prediction results.

FURTHER RESEARCH DIRECTIONS

The topic of stock prediction is of great interest. Other methods including Naive Baeys classification and Support Vector Machine may offer better, more accurate prediction models. It is interesting and worth investigating these approaches. A larger data set of at least 5 years should be used.

REFERENCES:

1. https://www.r-bloggers.com/illustrated-guide-to-roc-and-auc/
2. https://www.kaggle.com/aaron7sun/stocknews
3. http://stats.stackexchange.com/questions/132777/what-does-auc-stand-for-and-what-is-it
4. https://www.r-bloggers.com/illustrated-guide-to-roc-and-auc/
5. https://cran.r-project.org/web/packages/RWeka/RWeka.pdf