Bowen Zhao

Carolina Virreira

INEG 4163

March 17, 2023

YouTube Predictive Modeling

1. Introduction

YouTube is an online video-sharing platform that is well known for its collection of user-generated content. It enables users to upload, view, rate, share, and comment on a diverse range of videos that content creators share with users. One of the most important goals for content creators is to predict the growth rate of views for their newly uploaded videos. The objective of this analysis is to explain different statistical techniques/models used to come up with these predictions as accurately as possible.

2. Methodology

a) Preprocessing:

In order to achieve optimal results for our predictions, we first make sure our data is clean and ready to be analyzed. We started by creating a data frame containing only the predictor variables. We used filtering to remove non-informative or redundant predictors, in this case we deleted columns ID, Date and Growth. We decided to delete Date since it is not relevant on this analysis. We remove the response variable growth by separating it from the predictor variables. We analyze how the predictor variable affects the response variables without the response variable itself influencing the analysis. This helps us to identify which predictor variables are most strongly associated with the response variable and to develop a model that can accurately predict or explain the response variable.

Next, we discarded some predictors from the analysis using technique X. Predictors with near-zero variance have very little or no variation across them. We identified the predictors and removed them from the dataset. This avoids unnecessary variables that might give a less accurate prediction, as well as slowing down the model. This new data frame is our Predictor Matrix ready to be analyzed.

To finalize our data cleaning and transforming, we visualize our highly correlated predictors using a correlation matrix between the predictor variables. We identified the pairs of variables with a correlation greater than 0.75 and deleted them in order to avoid the problem of multicollinearity. Now we can mutate the response variable into the clean data to perform the rest of the statistical model.

b) Statistical Model:

Our first approach for our statistical was multiple linear regression model. The goal is to find a linear relationship between the predictors and response variables. We fitted a linear regression model having Growth as the response variable and the rest of the variables as predictor variables. With this model we can find different insights. We identified the predictor variables that are statistically significant and should be included in the reduced linear regression model. For example, the coefficient for "Duration" is negative, which means that as the duration of a video increases, the predicted growth actually decreases.

The results show that the multiple R-squared value of 0.5304 tells us that the model is explaining about 53% of the variability in the data, which means it is a moderately good fit. The "Adjusted R-squared" value of 0.5289 is similar, but it takes into account all of the variables in the model, which is important because adding more variables can often make the model fit better even if they don't add any predictive benefit. Comparing the reduced linear model with the first model, the multi liner full model had an R-squared value of 0.5304, while the reduced liner model had an R-squared value of 0.5214. The R-squared value indicates that the first model explains 53.04% of the variance in the target variable, while the second model explains 52.14%. However, the R-squared value does not change too much, but our group still needs to use the all-significant factors which p-value greater than 0.05.

Our second approach was to build a k-nearest neighbors (k-NN) regression model for predicting the Growth variable based on the other predictors. The core idea behind k-NN is to use the characteristics of the training dataset to make predictions for new data points. The model has the predictor variables centered and scaled with no resampling. The k-NN model is tuned for three different values of k: 1, 5, and 10. For k = 1 the score is 4.38802, k=5the score is 2.64765 and for k = 10 the score is 2.51928. Comparing the score of these three models we conclude that the model that outperforms the other models is k-NN with k = 10.

3. Results.

Our top score was 2.51928 using a k-NN model with k = 10, staying in the median of all the responses. From the results we can see that only 19 groups participated, and 5 groups left the competition. Those 5 groups were omitted. The average score for the top 19 results is 2.52060474. The minimum and maximum values are 2.39468 and 2.67311 which show that the average results were in an acceptable range. This also means that most of the models were following the correct approach since there are no outliers. We can also notice a trend regarding how many entries the group performed. The more entries, the better the score of the group.

4. Conclusions.

We believe that our model worked well because it used various techniques to preprocess the data and selected the relevant features for prediction. Correlations between the predictors were analyzed, built a linear regression model to predict the response variable and we trained k-nearest neighbors (k-NN) models with different values of and remove near-zero variance predictors. To further improve our model we could try different regression models and compare them to see which performance is best. We can also do more data preprocessing techniques to further understand the data.

5. Appendix.

~~~
```{r}
library(caret)
library(ggplot2)
library(corrplot)
library(dplyr)
library(readr)
```
```{r}
my_data <- training
```
```{r}
Predictor.my_data <- my_data[, -c(1,2,53)]
head(Predictor.my_data)
```
```{r}
zero <- nearZeroVar(Predictor.my_data)
print(zero)
```
```{r}
Predictor.Matrixc<- Predictor.my_data[, -zero]
head(Predictor.Matrixc)
```
```{r}
M <- cor(Predictor.Matrixc)
corrplot(M, method = "number")
~~~

~~~
findCorrelation(M, cutoff = 0.75)
```
```{r}
# Code for answer.
Predictor.Matrixca<- Predictor.Matrixc[, -c(13, 17, 14, 18, 20, 16, 11,  4,  6,  5, 10, 23)]
head(Predictor.Matrixca)
```
```{r}
# Code for answer
Corolla.Train <- mutate(Predictor.Matrixca, Growth=my_data$growth_2_6)
head(Corolla.Train)
names(Corolla.Train)
```
```{r}
mlr.model.full <- lm(Growth ~ ., data = Corolla.Train)
summary(mlr.model.full)
```
```{r}
mlr.model.reduced <- lm(Growth ~Duration+cnn_10+cnn_25 +cnn_68+  cnn_86
+edge_avg_value+num_words+num_stopwords+num_uppercase_chars

+num_uppercase_words
+num_digit_chars+Num_Subscribers_Base_low+Num_Subscribers_Base_low_mid+
Num_Views_Base_low+Num_Views_Base_low_mid+Num_Views_Base_mid_high
+count_vids_low_mid

, data = Corolla.Train)
summary(mlr.model.reduced)
```
```{r}
knn_control <- trainControl(method = "none")
m.knn.1 <- train(Growth ~ .,
~~~

~~~
                data = Corolla.Train,
                method = "knn",
                preProcess = c("center", "scale"),
                tuneGrid = data.frame(k = 1),
                trControl = knn_control)
```
```{r}
knn_control <- trainControl(method = "none")
m.knn.5 <- train(Growth ~ .,
                data = Corolla.Train,
                method = "knn",
                preProcess = c("center", "scale"),
                tuneGrid = data.frame(k = 5),
                 trControl = knn_control)
```
```{r}
knn_control <- trainControl(method = "none")
m.knn.10 <- train(Growth ~ .,
                data = Corolla.Train,
                method = "knn",
                preProcess = c("center", "scale"),
                tuneGrid = data.frame(k = 10),
                trControl = knn_control)
```
```{r}
knnPredict1 <- predict(m.knn.1, newdata = select(Corolla.Test, Duration
,views_2_hours,cnn_10,cnn_25,cnn_68,cnn_86,pct_nonzero_pixels,mean_red,mean_blue,
edge_avg_value,num_words,num_stopwords,num_uppercase_chars,num_uppercase_words,
num_digit_chars,Num_Subscribers_Base_low,Num_Subscribers_Base_low_mid,Num_Subscribers_Base_mid_high,
~~~

~~~
Num_Views_Base_low,Num_Views_Base_low_mid,Num_Views_Base_mid_high,count_vids_low_mid,

count_vids_mid_high,))

print(knnPredict1)

```

```{r}

mlr <- predict(mlr.model.reduced, newdata = Corolla.Test)

knnPredict1 <- predict(m.knn.1, newdata = select(Corolla.Test, Duration
,views_2_hours,cnn_10,cnn_25,cnn_68,cnn_86,pct_nonzero_pixels,mean_red,mean_blue,
edge_avg_value,num_words,num_stopwords,num_uppercase_chars,num_uppercase_words,

num_digit_chars,Num_Subscribers_Base_low,Num_Subscribers_Base_low_mid,Num_Subscribers_Base_mid_high,

Num_Views_Base_low,Num_Views_Base_low_mid,Num_Views_Base_mid_high,count_vids_low_mid,

count_vids_mid_high,))

output <- cbind(Corolla.Test[,1], knnPredict1)

colnames(output) <- c("id", "growth_2_6")

write.csv(output, file = "predictionskn1.csv", row.names = FALSE)

```

```{r}

knnPredict5 <- predict(m.knn.5, newdata = select(Corolla.Test, Duration
,views_2_hours,

cnn_10,                        cnn_25 ,

  cnn_68,                        cnn_86 ,

pct_nonzero_pixels,        mean_red,

mean_blue,                    edge_avg_value,

num_words,                    num_stopwords,

num_uppercase_chars,           num_uppercase_words,

num_digit_chars,              Num_Subscribers_Base_low,

Num_Subscribers_Base_low_mid,  Num_Subscribers_Base_mid_high,
~~~

~~~
Num_Views_Base_low,            Num_Views_Base_low_mid,
Num_Views_Base_mid_high,      count_vids_low_mid,
count_vids_mid_high,  ))
print(knnPredict1)
```
```{r}
mlr <- predict(mlr.model.reduced, newdata = Corolla.Test)
knnPredict1 <- predict(m.knn.1, newdata = select(Corolla.Test, Duration ,views_2_hours,cnn_10,cnn_25,cnn_68,cnn_86,pct_nonzero_pixels,mean_red,mean_blue, edge_avg_value,num_words,num_stopwords,num_uppercase_chars,num_uppercase_words,
num_digit_chars,Num_Subscribers_Base_low,Num_Subscribers_Base_low_mid,Num_Subscribers_Base_mid_high,
Num_Views_Base_low,Num_Views_Base_low_mid,Num_Views_Base_mid_high,count_vids_low_mid,
count_vids_mid_high,))
output <- cbind(Corolla.Test[,1], knnPredict5)
colnames(output) <- c("id", "growth_2_6")
write.csv(output, file = "predictionskn5.csv", row.names = FALSE)
```
```{r}
knnPredict10 <- predict(m.knn.10, newdata = select(Corolla.Test, Duration ,views_2_hours,
cnn_10,                       cnn_25 ,
  cnn_68,                       cnn_86 ,
pct_nonzero_pixels,        mean_red,
mean_blue,                    edge_avg_value,
num_words,                    num_stopwords,
num_uppercase_chars,           num_uppercase_words,
num_digit_chars,              Num_Subscribers_Base_low,
Num_Subscribers_Base_low_mid,  Num_Subscribers_Base_mid_high,
~~~

~~~
Num_Views_Base_low,            Num_Views_Base_low_mid,
Num_Views_Base_mid_high,      count_vids_low_mid,
count_vids_mid_high,  ))
print(knnPredict1)


```
```{r}
mlr <- predict(mlr.model.reduced, newdata = Corolla.Test)
knnPredict1 <- predict(m.knn.1, newdata = select(Corolla.Test, Duration
,views_2_hours,cnn_10,cnn_25,cnn_68,cnn_86,pct_nonzero_pixels,mean_red,mean_blue,
edge_avg_value,num_words,num_stopwords,num_uppercase_chars,num_uppercase_words,
num_digit_chars,Num_Subscribers_Base_low,Num_Subscribers_Base_low_mid,Num_Subscribers_Base_mid_high,
Num_Views_Base_low,Num_Views_Base_low_mid,Num_Views_Base_mid_high,count_vids_low_mid,
count_vids_mid_high,))
output <- cbind(Corolla.Test[,1], knnPredict10)
colnames(output) <- c("id", "growth_2_6")
write.csv(output, file = "predictionskn10.csv", row.names = FALSE)
`
~~~