MA5810

The Perfect Post for Facebook Fashion Sellers in Thailand

Assessment 3
CAPSTONE PROJECT

Bienvenido Hiyas Jr.
13824819

09-Dec-2020

The Perfect Post for Facebook Live Fashion Sellers in Thailand

Bienvenido Hiyas Jr. Student ID: 13824819

Abstract— **Online selling is common nowadays especially in social media like Instagram, twitter and Facebook. However, there is no guarantee that every time you post or sell your products every day, people will be interested in it. Furthermore, it is not clear what type of post should be done by the online sellers to gather different views or reactions. This report will provide a prediction model that will identify the type of post (Video, link, Photo and Status) of a data basing on the different predictors; number of reactions i.e likes, loves, wows, hahas, sads, and angrys; number for comments and number of shares. This report will also try to classify and find some interesting patterns of the different status type basing on the number of engagements like reactions, comments and shares. Both supervised and unsupervised learning was used in the data set in order come up with the result. Outliers were identified using KNN Outliers algorithm, Principal Component Analysis was also used to determine the principal components of the data, then K-means for clustering and Naïve Bayes for prediction. The model that was implemented on this report has more than 70% accuracy. With this, online sellers when to focus their online selling activity.**

I. Introduction

Selling on the Thai ecommerce market is a great option for growth-minded online sellers according to the Web Interpret website (www.webinterpret.com). In addition, there are 29,078,158 internet users in Thailand and Internet penetration amounts to 42.70%. One of the most common internet platform used in Thailand is Facebook. Thailand has 46 million registered Facebook users. Therefore, of all the Facebook accounts in the world, 2% of them are logging on from Thailand and this number represents a huge percentage of the Thai population and clearly Facebook is an essential part of daily life for most Thai people [1].

With this, most Thai fashion and cosmetics retail sellers are actively posting their products in Facebook via videos, pictures, statuses and links and gathering responses like comments, sharing and reactions. However, there seems to be a significant difference of Facebook users engagements in every quarter and months within the year with regards to their post. This report will provide a quarterly and monthly statistical analysis of the Facebook posts from 10 Thai fashion and cosmetic retail sellers as well as to provide a visual presentation of the comments, shares and reactions as well as the type of posts that gathered the highest numbers of reactions, comments and shares.

II. Data

The data set used in this investigative analysis is the Facebook Live Sellers in Thailand Data Set, from Nassim Dehouche, Mahidol University International College, nassim.deh '@' mahidol.edu.

The data file is a comma separated values file named, Live.csv [2] and the Data Set Description can be found in UCI Machine Learning Repository [3]. The Live.csv file has 7051 rows and 16 Columns. However, the last 4 columns named Column1, Column2, Column3 and Column4 don't have values data on each column. The first row contains the variables indicated on table 1 below.

Variables	Type
status_id	Qualitative Nominal
status_type	Qualitative Nominal
status_published	Qualitative Ordinal
num_reactions	Quantitative Discrete
num_comments	Quantitative Discrete
num_shares	Quantitative Discrete
num_likes	Quantitative Discrete
num_loves	Quantitative Discrete
num_wows	Quantitative Discrete
num_hahas	Quantitative Discrete
num_sads	Quantitative Discrete
num_angrys	Quantitative Discrete
Column1	N/A
Column2	N/A
Column3	N/A
Column4	N/A

Table1 – List of Variables and Types.

The data set was taken from Facebook pages of 10 Thai fashion and cosmetics retail sellers posts of a different nature **status_type** (video, photos, statuses, and links). The engagement metrics consist of comments, shares, and reactions. Moreover, the number of reactions is equivaent to the total number of likes, loves, wows, hahas, sads and angrys.

Prior to these investigation, the variability of consumer engagement is analysed through a Principal Component Analysis, highlighting the changes induced by the use of Facebook Live. The seasonal component is analysed through a study of the averages of the different engagement metrics for different time-frames (hourly, daily and monthly). Finally, the statistical outlier posts were identified, that are qualitatively analysed further, in terms of their selling approach and activities.

III. Methods

A range of Data Mining methods with R studio commands and libraries are used for the pre-processing of the Live.csv data. These were conducted in the R-Studio software[4]. The libraries used are ISLR, caret, dplyr, cluster, factoextra, psych and dbscan

A. Data Presentation

The R function **read.csv()** was used to import the Live.csv data to R studio. Then, the argument **header** was set to **T**(True) to instruct R that the first line of the file are the variable names. The new data is saved in the new data named, **data**. **data** has now a total of 7050 observations with 16 variables. **Data** was then inspected with the **summary()** command and found that there are no missing values in every observation except for the last 4 variables; **Column1, Column2, Column3 and Column4** where they don't have data or values. Therefore in order to eliminate them, the **data <- data[,1:12]** command was used excluding the unnecessary four variables mentioned above.

B. Type Conversion

When the data is imported using the function **read.csv()**, the variable type is automatically assigned. To perform some types of numerical analysis, the data types of these variables were manually assigned. The variables **status_id** and **status_type** factors were as char therefore they are converted to factor variables using **as.factor() command**. The other variables were already assigned as **int,** which is already enough. Status_published variable was not transformed since this variable will be discarded or will not be used.

Then to identify duplicate rows the command **which(duplicated(data))** was implemented and then delete it from the data using **data <- data[-which(duplicated(data)),]** command. Now the observations are 6999.

In result to above data presentation and type conversion methods, the data is now ready for data mining.

C. KNN Outlier Detection

KNN method uses all data to be numeric/int therefore the predictors were selected and stored to new variable **data_Outlier** using the command **data_Outlier <- data[-c(1:3)].** Then the KNN parameter was set to 4 and the top 20 outliers were selected. The rank of the outliers were also set using the **order()** command. The KNN_Result was done using the command data.frame() with the ID set as the rank and the score as the KNN_Outlier. To view the result with the top 20 outliers. The command **head(KNN_Result, top_n)** was implemented.

To plot the result, ggplot() and geom_point() was used using the data_Outlier with the aes mapping of num_reaction and num shares.

Now that we have identified the outliers, we can now delete them from our data set and stored in a new object **data_final** using the command.
data_final <- data[-c(499,481,6707,4544,3247,4515,3893,727,6777,6609,6712,4519,6725,6749,4612, 6246,3852,4563, 3877,1230),]
(Note that these are the row numbers of the Outliers)

Now our data_Final has 6979 obervations.

D. PCA (Principal Component Analysis)

PCA method needs to have numerical variables and the row name must be the status_type. Therefore, the following commands was implemented.

#identify duplicate rows
which(duplicated(data_PCA))
#delete duplicated rows that are identified
data_PCA <- data_PCA[-which(duplicated(data_PCA)),] #now observations are 6977
#make status_id row names
row.names(data_PCA) <- data_PCA$status_id
#delete status_id columns
data_PCA <-data_PCA[,-1]

PCA algorithm was then implemented using the function **prcomp()** with the paramenters center and scale were set to **TRUE**

Finally to visualize the result of the PCA, the command **plot()** was used as well as the **fviz_screplot().**

Now that we identified the principal component in the PCA analysis, our data is now reduced to 3 variables and stored to object data_final1 using the command **data_final1 <- data_PCA[c(1:3)]**

E. K-means Algorithm (Clustering)

The data used for K-means was the same data as the **data_final1** and stored to new objefct **data_k.** **data_k** is then scaled using **scale()**.

K-means was then implemented using **kmeans()** command with data_k and set the parameters centers to 4 and nstart to 25. The tot.withinss result can be seen usitn the **str()** command.

To visualize the k-means result, the **fviz_cluster()** command was implemented with the data data_k, geom = "point", shape = 19, alpha = 0. Geom_point was also added with aes colour set as the **kmeans_data$cluster** and the shape = **data_final1$status_type.**

To perform kmeans & calculate the SSE the commands below were implemented.

#perform kmeans & calculate ss
total_sum_squares <- function(k){
kmeans(data_k, centers = k, nstart = 25)$tot.withinss
}

#define a sequence of values for k up to 10 sequences
all_ks <- seq(1,10,1)

#apply to all values of k
choose_k <- sapply(seq_along(all_ks), function(i){
total_sum_squares(all_ks[i])
})
 # dataframe for plotting
choose_k_plot <- data.frame(k = all_ks,
within_cluster_variation = choose_k)

To visualize the Number of clusters and within cluster variation, ggplot() was using the choose_k_plot along with geom_point() and geom_line().

F. Naïve Bayes / Regression

To check if the predictors are independent variables are correlated, the function **cor()** was used with the argument **method="pearson"** to measure the linear

correlation between the numeric variables using the Pearson Correlation method.
and a visualization was implemented using **pairs.panels()** on the data_final.

The data was then reduced to the dependent variable and the predictors using the commands
data_final <- data_final[c(1:6)]
data_final <- data_final[,-c(1,3)]

Then the Naïve bayes algorithm was implemented to predict our dependent variable with the predictors. The commands below were implemented.

```
##partition data set into train and set and randomly split it
set.seed(123)
# Creating index for randomly splitting the dataset
ind3= createDataPartition(data_final$status_type, p=0.8, list=FALSE)
train.data <- data_final[ind3,]
test.data <- data_final[-ind3,]
# Training the model
c(nrow(train.data), nrow(test.data))
summary(data_final)
#implementing Naive Bayes
model <- train(status_type~.,
        data=train.data,
        trControl = trainControl(method = "cv", number = 5),
        method = "nb")
model
#confusion Matrix for training data
confusionMatrix(predict(model,newdata = train.data),
        train.data$status_type)
#confusion Matrix for test data
confusionMatrix(predict(model,newdata = test.data),
        test.data$status_type)
```

IV. Results and Discussions

The original data from the Facebooksellers.csv file was spreadsheet of 7051 rows and 16 columns. After doing the data presentation, the result is a data set of 7050 observation and 12 variables where the first row of the data as the variables and the unnecessary columns were omitted. After doing data transformation, elimination of duplicate rows our data now has 6999 observations

KNN Outliers Detection algorithm showed the top 20 outliers as shown in Figure 1 below.

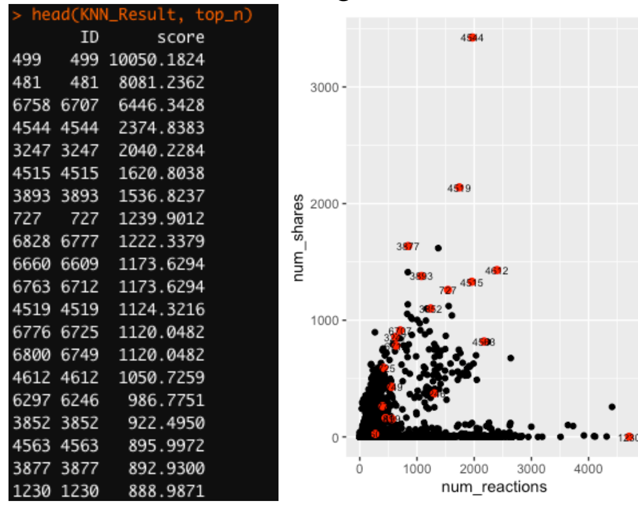
```
> head(KNN_Result, top_n)
       ID      score
499   499 10050.1824
481   481  8081.2362
6758 6707  6446.3428
4544 4544  2374.8383
3247 3247  2040.2284
4515 4515  1620.8038
3893 3893  1536.8237
727   727  1239.9012
6828 6777  1222.3379
6660 6609  1173.6294
6763 6712  1173.6294
4519 4519  1124.3216
6776 6725  1120.0482
6800 6749  1120.0482
4612 4612  1050.7259
6297 6246   986.7751
3852 3852   922.4950
4563 4563   895.9972
3877 3877   892.9300
1230 1230   888.9871
```

Figure 1 – Top 20 Outliers from KNN Outlier Detection and ggplot

This Outliers can have an impact to PCA algorithm therefore these were eliminated in **data_final.**

PCA algorithm results shows the dimensions and principal components as shown in Figure 2 below.

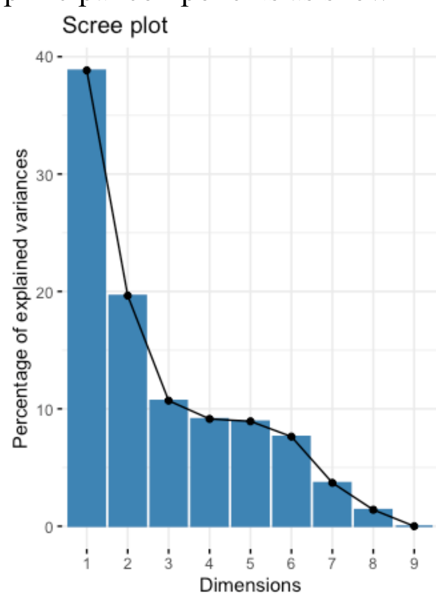

Figure 2 – Scree Plot on PCA result

Figure 2 shows the amount percentage of variance that explain by each principal component axis. It explains 40 percent of the variation for dimension 1 within the total data set. In the image shown above sharp bend is at 3. So, the number of principal axes should be 3. This is consistent on the variables since the num_reactions is the total number of num_likes, num_wows, num_hahas, num_sands and num_angrys. Thus, we can eliminate either of them. In our result, we will retain num_reactions, num_comments and num_shares only.

K-means results compares the clustering of the status type (link, photo, status, video) using K = 4. It can be noticed that videos (+) were properly clustered.

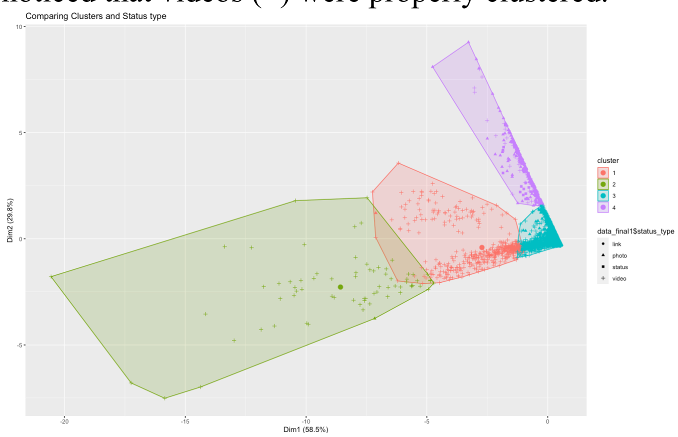

Figure 3 – Comparing clusters and Status Type

Naïve Bayes Prediction

First, the correlation matrix was implemented to verify the correlation of the predictors. The result can be found in the figure 4 below.

	num_reactions	num_comments	num_shares
num_reactions	1.0000000	0.1561897	0.2596399
num_comments	0.1561897	1.0000000	0.6405356
num_shares	0.2596399	0.6405356	1.0000000

Figure 4 – Pearson Correlation Result for predictors.

In addition, a data visualization on the relationship in figure 1 above was also done to have a clear visualization as shown below Figure 5.

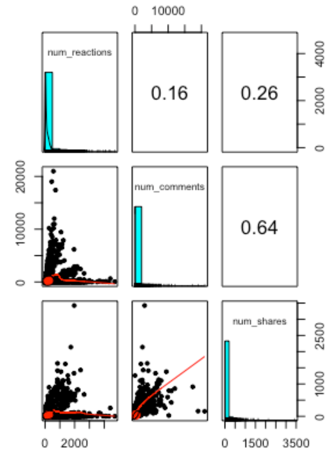

Figure 5 – Visualization and correlation result

Figure 5 shows that the predictors are not normal distribution therefore, LDA and QDA cannot be used and correlation result shows less correlation between predictors, therefore, Naïve Bayes can be implemented.

Naïve Bayes model results shows 72% accuracy for the training data as shown in Figure 6 below.

```
Confusion Matrix and Statistics

          Reference
Prediction link photo status video
    link      0     0      0     2
    photo    50  3364    279  1101
    status    1     0      3    13
    video     0    32      6   751

Overall Statistics

               Accuracy : 0.7351
                 95% CI : (0.7233, 0.7466)
```

Figure 6 – Confusion matrix result for Training data.

Naïve Bayes model results shows 74% for the test data as shown in Figure 7 below.

```
Confusion Matrix and Statistics

          Reference
Prediction link photo status video
    link      0     0      0     2
    photo    12   833     70   262
    status    0     2      0     4
    video     0    13      1   198

Overall Statistics

               Accuracy : 0.738
                 95% CI : (0.7141, 0.7609)
```

Figure 7 – Confusion matrix result for Test data.

This indicates that our model can have a good prediction for the Type of Status to be used by online Sellers.

V. Conclusion

This investigation has successfully achieved its objective which is to provide a model to predict the type of post (Video, Link, Photo, Status) basing on the number of reactions, shares and comments of the Facebook posts from 10 Thai fashion and cosmetics retail sellers with the accuracy more than 70%. The statys types were also classified correctly using k=4 the different types of post using k-means algorithm

Although this investigation is only limited to the data provided from 2013 to 2017 of the 10 Thai fashion and Cosmetic retail sellers in Facebook, this can also be used as a base platform for further investigation specially for the latest years where online selling activity is still popular in Thailand. Further data and predictors can also be added in order to improve the accuracy of the model.

VI. Preferences

[1] "Who are Thailand's 46 Million Facebook Users?" Retrieved August 2, 2017 from https://www.bangkokpost.com/learning/learning-together/1296218/who-are-thailands-46-million-facebook-users-

[2] Index of /ml/machine-learning-databases/00488. Retrieved from https://archive.ics.uci.edu/ml/machine-learning-databases/00488/

[3] Facebook Live Sellers in Thailand Data Set. Retrieved from https://archive.ics.uci.edu/ml/datasets/Facebook+Live+Sellers+in+Thailand

[4] RStudio Team (2020). RStudio: Integrated Development Environment for R. RStudio, PBC, Boston, MA URL http://www.rstudio.com/. Mode: Desktop. Version: 1.3.1093. Release Name: Apricot Nasturtium

VII. Appendix

R. Code used for this Data Mining.

```
rm(list=ls())

library(ISLR)
library(caret, warn.conflicts = F, quietly = T)
library(dplyr)
library(cluster, warn.conflicts = F, quietly = T)
#clustering algorithms
library(factoextra, warn.conflicts = F, quietly = T)
#data visualization
library(psych)
library(dbscan)


#Clustering
#capstone
data <- read.csv("FacebookSellers.csv", header =
TRUE)
summary(data)
str(data)
#data Transformation
data$status_id <-as.factor(data$status_id)
data$status_type <-as.factor(data$status_type)
#delete NA columns
data <- data[,1:12]
data <- na.omit(data)

#identify duplicate rows
which(duplicated(data))
#delete duplicated rows that are identified
data <- data[-which(duplicated(data)),] #now
observations are 6999

#### KNN Outlier
#install.packages("dbscan")
```

```
data_Outlier <- data[-c(1:3)]
#KNN parameter
k=4
#using latest version for KNN (All - TRUE)
KNN_Outlier <- kNNdist(x=data_Outlier, k =k, all
= TRUE)[,k]
#No. of top outliers to be displayed
top_n <- 20
#sorting Outliers
rank_KNN_Outlier <- order(KNN_Outlier,
decreasing = TRUE)
KNN_Result <- data.frame(ID =
rank_KNN_Outlier, score =
KNN_Outlier[rank_KNN_Outlier])
#showing top 20 outliers with ID's and scores
head(KNN_Result, top_n)

#plotting the Outliers
g0a <- ggplot() + geom_point(data=data_Outlier,
mapping=aes(x=num_reactions, y= num_shares),
shape = 19)
g <- g0a +
 geom_point(data =
data_Outlier[rank_KNN_Outlier[1:top_n],],
mapping = aes(x=num_reactions,y=num_shares),
shape=19, color="red", size=2) +

geom_text(data=data_Outlier[rank_KNN_Outlier[1
:top_n],],
        mapping=aes(x=(num_reactions-0.5),
y=num_shares, label=rank_KNN_Outlier[1:top_n]),
size=2.5)
g

#delete Outliers from for final data
data_final <- data[-
c(499,481,6707,4544,3247,4515,3893,727,6777,66
09,6712,4519,6725,6749,4612,
                6246,3852,4563, 3877,1230),]

#PCA
#show how samples are related or not to each other
#delete unwanted variables/data

#delete status_type and status_published variables
for PCA
data_PCA <- data_final[-c(2:3)]
#identify duplicate rows
which(duplicated(data_PCA))
#delete duplicated rows that are identified
data_PCA <- data_PCA[-
which(duplicated(data_PCA)),] #now observations
are 6977

#make status_id row names
row.names(data_PCA) <- data_PCA$status_id
#delete status_id columns
data_PCA <-data_PCA[,-1]

#perform PCA
pca_res <- prcomp(data_PCA, center = TRUE, scale
= TRUE)
pca_res
pca_res$rotation
pca_res$x
#plot
plot(pca_res$x[,1], pca_res$x[,2])

fviz_screeplot(pca_res)
#shows the amount percentage of variance that
explain by each principal component axis
#explains 60percent of the variation within the total
data set.
#For example in the image shown above sharp bend
is at 3. So, the number of principal axes should be 3.


data_final1 <- data[-c(1,3)]
data_final1 <- data_final1[c(1:4)]
###

#data_K is the data with only 3 variables as result
from PCA
data_k <- data_final1[,-1]
#scaling data_K to normalise data
data_k <- scale(data_k)


set.seed(6)
#Appply K means algorithm
kmeans_data <- kmeans(data_k, centers = 4, nstart =
10)
str(kmeans_data)
fviz_cluster(kmeans_data, data = data_k)
```

```
fviz_cluster(kmeans_data, #set up plot
        data = data_k,
        geom = "point", # only shows points and not labels
        shape = 19,# define one shape for all clusters (a circle)
        alpha = 0)+ # make circles see-though
  geom_point(aes(colour = as.factor(kmeans_data$cluster),
          shape = data_final1$status_type))+ #colour by status type
  ggtitle("Comparing Clusters and Status type") #add a title

#perform kmeans & calculate ss
total_sum_squares <- function(k){
  kmeans(data_k, centers = k, nstart = 10)$tot.withinss
}

#define a sequence of values for k up to 10 sequences
all_ks <- seq(1,10,1)

#apply to all values of k
choose_k <- sapply(seq_along(all_ks), function(i){
  total_sum_squares(all_ks[i])
})
# dataframe for plotting
choose_k_plot <- data.frame(k = all_ks,
              within_cluster_variation = choose_k)
# plot
ggplot(choose_k_plot, aes(x = k,
             y = within_cluster_variation))+
  geom_point()+
  geom_line()+
  xlab("Number of Clusters (K)")+
  ylab("Within Cluster Variation")


#Supervised Learning/ Regression

data_model <- data_final1[,-1]
#checking for predictors correlation
cor_data <- cor(data_model,method = "pearson")
cor_data
pairs.panels(data_model)
#Correlation shows low correlation between variables therefore assume that the yare independent, Hence Naive Bayes will be use for prediction


##partition data set into train and set and randomly split it
set.seed(123)
# Creating index for randomly splitting the dataset
ind3= createDataPartition(data_final$status_type, p=0.8, list=FALSE)
train.data <- data_final[ind3,]
test.data <- data_final[-ind3,]
# Training the model
c(nrow(train.data), nrow(test.data))
summary(data_final)
#implementing Naive Bayes
model <- train(status_type~.,
        data=train.data,
        trControl = trainControl(method = "cv", number = 5),
        method = "nb")
model
#confusion Matrix for training data
confusionMatrix(predict(model,newdata = train.data),
         train.data$status_type)
#confusion Matrix for test data
confusionMatrix(predict(model,newdata = test.data),
         test.data$status_type)
```