Untitled

Soham Neeraj Agarkar (1002157894)

2024-05-07

```
{r setup, include=FALSE} knitr::opts_chunk$set(echo = TRUE)
```

Social Media Activity Analysis using K-means and K-Medioids Clustering

-Soham Neeraj Agarkar & Huakang Lu

Abstract

This project aims to make use of the Facebook-Live-Sellers Dataset acquired from UCI machine Learning Repository. It is a CSV Dataset consisting of 7,050 Facebook posts of various types (text, deferred and live videos , images). These posts were extracted from the Facebook pages of 10 Thai fashion and cosmetics and retail sellers from March 2012 - June 2018. The dataset was collected via the Facebook API, and anonymized in compliance with the Facebook Platfrom Policy for Developers. For each Facebook post, the dataset records the resulting engagement metrics comprising shares, comments, and emoji reactions within which we distinguish traditional "likes" from recently introduced emoji reactions, that are "love", "wow", "haha", "sad" and "angry".

The Goal of the project is to perfrom K-means and K-Medioids classifications on the data and derive suittable insights as well as compare the results of clustering methods.

EDA

Libraries

```
library(cowplot)
library(ggcorrplot)
library(ggplot2)
library(stats)
library(cluster)
library(GGally)
library(kableExtra)
library(factoextra)
library(tidyverse)
library(tinytex)
```

Loading the Dataset

```
#Load the dataset
df <- read.csv("C:/Users/soham/Documents/Github rep/6303-Final-
Project/facebook+live+sellers+in+thailand/Live_20210128.csv",
row.names= NULL)
```

Feature Description

Variable Name	Role	Type	Missing Values
status_id	ID	Integer	No
status_type	Feature	Categorical	No
status_published	Feature	Categorical	No
num_reactions	Feature	Integer	No
num_comments	Feature	Integer	No
num_shares	Feature	Binary	No
num_likes	Feature	Integer	No
num_loves	Feature	Binary	No
num_wows	Feature	Binary	No
num_hahas	Feature	Binary	No
num_sads	Feature	Binary	No
num_angrys	Feature	Binary	No

Data View

A small sample of the Data was viewed to see some details such as available features, the various +-datatypes comprised within the dataset as well its Quartiles.

```
#data view
head(df)

str(df)

summary(df)
```

Plots

```
#Plots
df$status_type <- factor(df$status_type, levels = c("link", "photo",
"status", "video"), labels = c(1, 2, 3, 4))

# Histogram of num_reactions
plot1 <- ggplot(df, aes(x = num_reactions)) +
  geom_histogram(binwidth = 20, fill = "skyblue", color = "black") +
  labs(title = "Distribution of Number of Reactions")

# Histogram of num_comments
plot2 <- ggplot(df, aes(x = num_comments)) +
  geom_histogram(binwidth = 20, fill = "skyblue", color = "black") +
  labs(title = "Distribution of Number of Comments")
```

```
# Histogram of num_shares
plot3 <- ggplot(df, aes(x = num_shares)) +
  geom_histogram(binwidth = 20, fill = "skyblue", color = "black") +
  labs(title = "Distribution of Number of Shares")

# Histogram of num_likes
plot4 <- ggplot(df, aes(x = num_likes)) +
  geom_histogram(binwidth = 20, fill = "skyblue", color = "black") +
  labs(title = "Distribution of Number of Likes")

# Histogram of num_loves
plot5 <- ggplot(df, aes(x = num_loves)) +
  geom_histogram(binwidth = 20, fill = "skyblue", color = "black") +
  labs(title = "Distribution of Number of Loves")

# Histogram of num_wows
plot6 <- ggplot(df, aes(x = num_wows)) +
  geom_histogram(binwidth = 20, fill = "skyblue", color = "black") +
  labs(title = "Distribution of Number of Wows")

# Histogram of num_hahas
plot7 <- ggplot(df, aes(x = num_hahas)) +
  geom_histogram(binwidth = 20, fill = "skyblue", color = "black") +
  labs(title = "Distribution of Number of Hahas")

# Histogram of num_sads
plot8 <- ggplot(df, aes(x = num_sads)) +
  geom_histogram(binwidth = 20, fill = "skyblue", color = "black") +
  labs(title = "Distribution of Number of Sads")

# Histogram of num_angrys
plot9 <- ggplot(df, aes(x = num_angrys)) +
  geom_histogram(binwidth = 20, fill = "skyblue", color = "black") +
  labs(title = "Distribution of Number of Angrys")

# Arrange plots in a grid
plot_grid(plot1, plot2, plot3, plot4, plot5, plot6, plot7, plot8,
plot9, ncol = 3)

#Grouped bar plot for comparison between status_type
# Melt the data frame to long format
data_long <- reshape2::melt(df, id.vars = "status_type")

# Create a grouped bar plot
ggplot(data_long, aes(x = variable, y = value, fill = status_type)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Comparison of Features for Photo and Video Types",
       x = "Feature",
       y = "Value") +
  facet_wrap(~ variable, scales = "free_y") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

#Drop status_published, status_type
data_subset <- select(df, -c("status_published","status_type"))

#scatterplot matrix
pairs(data_subset)
```

As part of the EDA, we looked at the above plots. A common trend that we saw is that most reactions tend to amount to 0, i.e, the post didn't receive said reaction. This makes sense given that it isn't unusual for a person to provide any more than one like or one comment as part of interaction with the post.

Looking at the scatterplot matrix, it is clear that num_reactions has a highly positive correlation with num_likes.

Correlation

Having looked at the scatterplot matrix, we plotted the correlation matrix.

```
#calculate correlation matrix
correlation_matrix <- cor(select(df, -
c("status_type","status_published")))

#Plot correlation matrix
ggcorrplot(correlation_matrix, type = "lower", lab = TRUE)
```

The correlation only confirmed our findings. num_reactions has an almost perfect positive correlation with num_likes. This indicates that the increasing number of reactions on the post were directly tied to the number of likes the post was receiving. The other takeaway here is that majority of the reactions to the post had to do with liking the post itself.

Modelling and Silhouette

Firstly, we seperate the features with the highest correlations and scale the the new dataframe.

```
#considering num_likes, num_reactions, numa_shares, num_comments and
num_loves as a datset due to high correlation
new_df <- select(data_subset, c("num_reactions", "num_likes",
"num_shares",
                                "num_comments", "num_loves"))

#Scaling the Data
data_subset_scaled <- scale(new_df)
```

K-Means

The first step in our modeling was acquiring the optimal amount of clusters to be used.

```
#Modelling
#Kmeans
#Acquiirng the the optimal number of clusters for the model using
scree plot
fviz_nbclust(data_subset_scaled, kmeans, method="wss") +
  geom_vline(xintercept = 4, linetype = 2)
```

The plot above shows the optimal number of clusters that should be used in our model.

```
#Kmeans model
set.seed(7894)
kmeans_model <- kmeans(data_subset_scaled, nstart=20, centers=4)
print(kmeans_model)

data_subset$cluster <- kmeans_model$cluster


pairs(data_subset[, -ncol(data_subset)],
      col = kmeans_model$cluster,
      pch = 16,
      main = "Scatterplot Matrix with Clusters")
```

Above is a scatter plot matrix of all the interactions with the posts color coded by the clusters within which they are present.

```
#Aggregating the clusters
kable(aggregate(data_subset, by=list(cluster=kmeans_model$cluster),
mean),
      format = "latex",
      booktabs = TRUE) %>% kable_styling(position="center")
```

This is another visualization for the scatterplot matrix from earlier that shows the clustering more clearly.

```
#cluster visualization
fviz_cluster(kmeans_model, data_subset)

# Calculate silhouette scores
silhouette_scores <- silhouette(kmeans_model$cluster,
dist(data_subset_scaled))


# Mean silhouette score
mean_silhouette_score <- mean(silhouette_scores[, "sil_width"])
print(mean_silhouette_score)
```

Our K-means model received a silhouette score of **0.7578**.

This suggests that the clusters are well-separated and that each data point is relatively close to its own cluster centroid compared to other clusters. This indicates a strong clustering structure in our data, where the clusters are distinct and well-defined.

K-Medioids

Much like K-Means, we derive the optimal number of clusters for the K-Medioids model

```
#Kmedioids
fviz_nbclust(data_subset_scaled, pam, method="wss") +
  geom_vline(xintercept = 5, linetype = 2)

set.seed(7894)
kmed_model <- pam(data_subset_scaled, k=5)
print(kmed_model)

pairs(data_subset_scaled[, -ncol(data_subset_scaled)],
     col = kmed_model$cluster,
     pch = 16,
     main = "Scatterplot Matrix with Clusters")

kable(aggregate(data_subset, by=list(cluster=kmeans_model$cluster), 
mean),
      format = "latex",
      booktabs = TRUE) %>% kable_styling(position="center")

#cluster plot
fviz_cluster(kmed_model, data_subset_scaled)

#silhouette socres
sh_scores <- silhouette(kmed_model$cluster, dist(data_subset_scaled))


#Mean silhouette scores
mean_sh_score <- mean(sh_scores[, "sil_width"])
print(mean_sh_score)
```

The model received a silhouette score of **0.62178**.

Much like K-means, this suggests that the model is a decent fit as the points are well-matched to the cluster in majority of the cases.

Conclusion

Upon performing K-means and K-Medioids classification on the data, we can conclude that K-means is a better classifier. This is indicated by the Silhouette Score, where K-means had a better score suggesting that is capable of more successfully classifying the reactions given to each post.