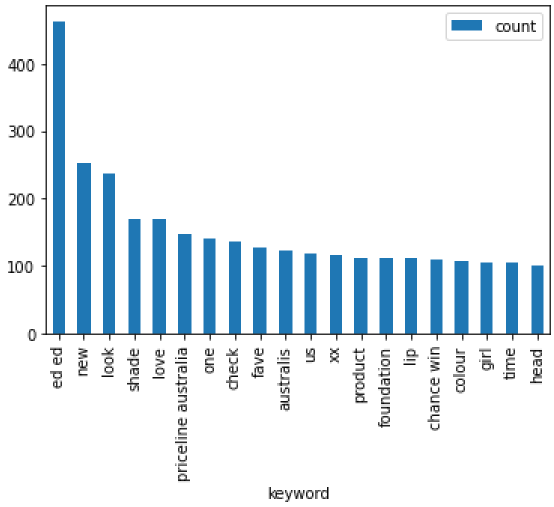
Facebook page activity tracking | "Australis Cosmetics" Company

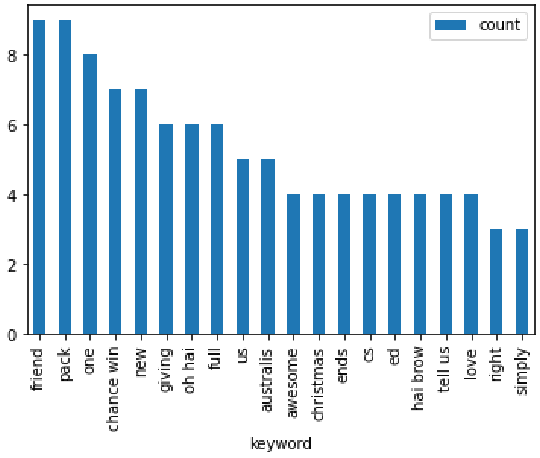
Instructors: Beau Giannini, Pavel Paramonov
Ohyoung Kim

The central challenge in establishing a Facebook brand page and in using it effectively to reach a company's marketing objectives lies in stimulating consumer engagement (Int. J. Market Research, 2018). When you post posts on Facebook, your audience get the opportunity to 'like', 'comment' or 'share'. Especially, 'share' is a much more forceful action, enabling the reader to push your post out to their friends and followers in a manner which says, 'I totally agree with what this post is saying' (2021).

This report aims to analyze messages based on the entire dataset and *shares_count*, suggesting how they are currently being used and how Facebook should be used to increase *shares_count* in the future. It uses a representative dataset was collected from the public page of an Australian cosmetics company "Australis Cosmetics" over the period between 2012 and 2017. There is no API endpoint to describe since the dataset was already pre-assembled.

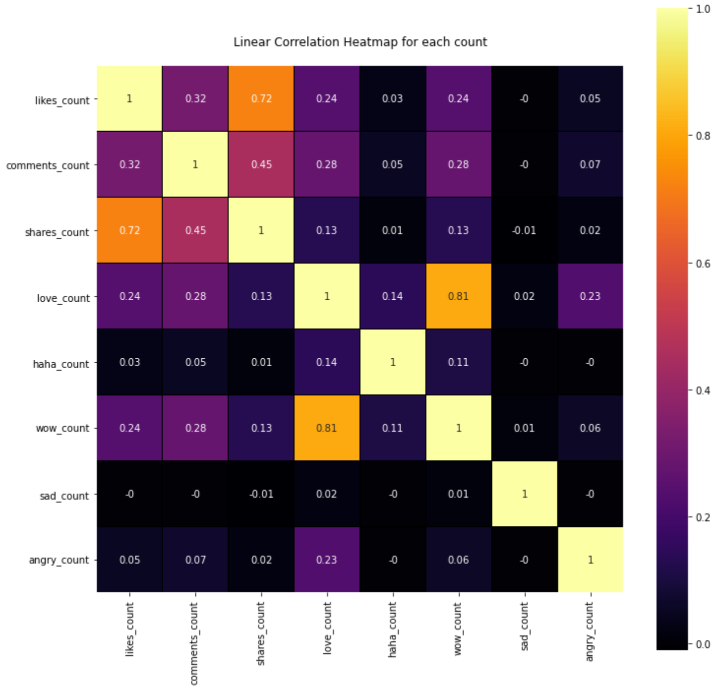
Of the four content types identified in the dataset, *photo* is the most popular and shared content type on this platform, with 3.89 times on average. Since the other three types—*video*, *link*, and *status*—have been shared 0 times on average, Australis Cosmetics should make its content into a *photo* type.





Analyzing the overall message (left side) shows that the word '*new*' has been used a lot. This showed that the company was using Facebook to promote the release of its products. Words such as '*chance win*' showed that the company was also using Facebook for marketing campaigns. However, a comparative analysis of the count of words '*new*' and '*chance win*' mentioned in the entire message and in the shared messages (right side) suggests that the company should use Facebook as a campaign space rather than a space to promote new products. Also, messages based on the count of shares mentioned the "*Oh Hai Brow*" product (which is identified in the form of "*oh hai*" and "*hai brow*" in bar chart), while the overall message did not identify the words. Since the interest in this product seems to be high, I suggest planning a related campaign. Keyword is ‘*love*’.

Best time to post is between 4 PM and 6 PM, with the most shared posts posted by 5 PM. 10 AM is also a good time. Therefore, it is suggested to post content at 10 AM and watch the situation until 5 PM, or post content at 4 PM, check the response by the next morning, and decide how to manage the content.



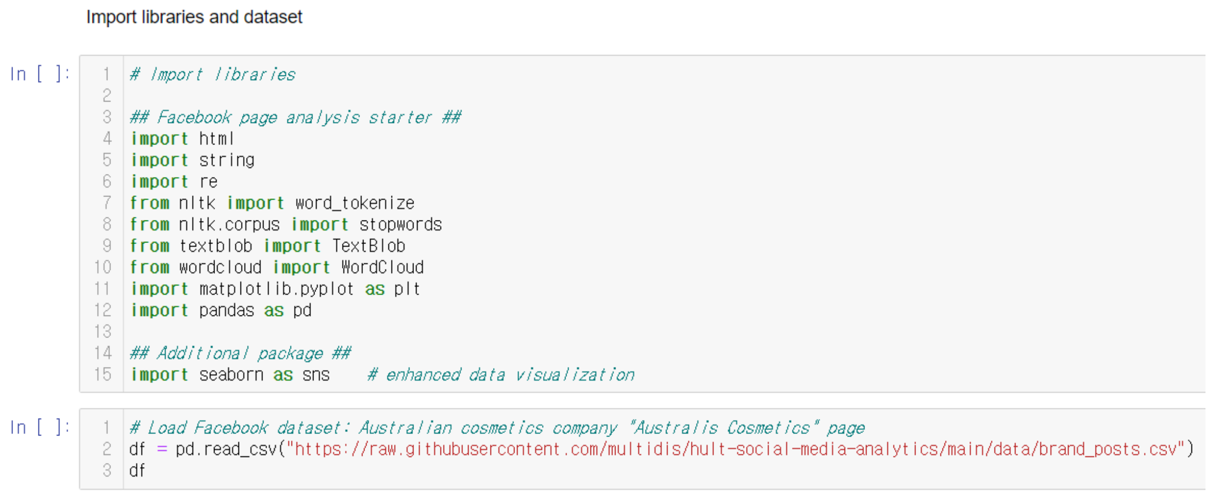
Finally, *shares_count* seems relatively related to *love_count* and *wow_count*. However, these two emotion counts, unlike the absolutely positive *haha_count*, are neutral feelings related to angry which is generally considered negative, so you should be careful not to be considered negative to consumers when creating content using it.

REFERENCE

Haikel-Elsabeh, M., Zhao, Z., Ivens, B., & Brem, A. (2018). *When is brand content shared on Facebook? A field study on online Word-of-Mouth*. International Journal of Market Research, 61(3), 287-301. doi:10.1177/1470785318805301

How do Likes, Comments and Shares Affect the Visibility of your Posts on your Facebook Page? (2018, January 24). Retrieved July 16, 2021, from https://www.bigfootdigital.co.uk/likes-comments-shares-affect-visibility-posts-facebook-page

APPENDIX



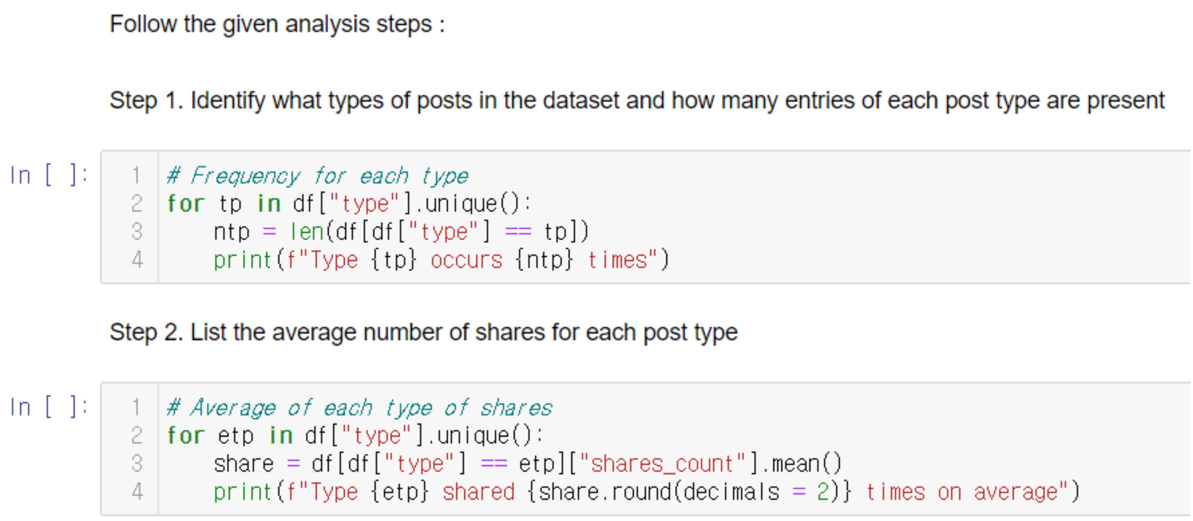
Import libraries and dataset

In [ ]:
```
# Import libraries

## Facebook page analysis starter ##
import html
import string
import re
from nltk import word_tokenize
from nltk.corpus import stopwords
from textblob import TextBlob
from wordcloud import WordCloud
import matplotlib.pyplot as plt
import pandas as pd

## Additional package ##
import seaborn as sns    # enhanced data visualization
```

In [ ]:
```
# Load Facebook dataset: Australian cosmetics company "Australis Cosmetics" page
df = pd.read_csv("https://raw.githubusercontent.com/multidis/hult-social-media-analytics/main/data/brand_posts.csv")
df
```



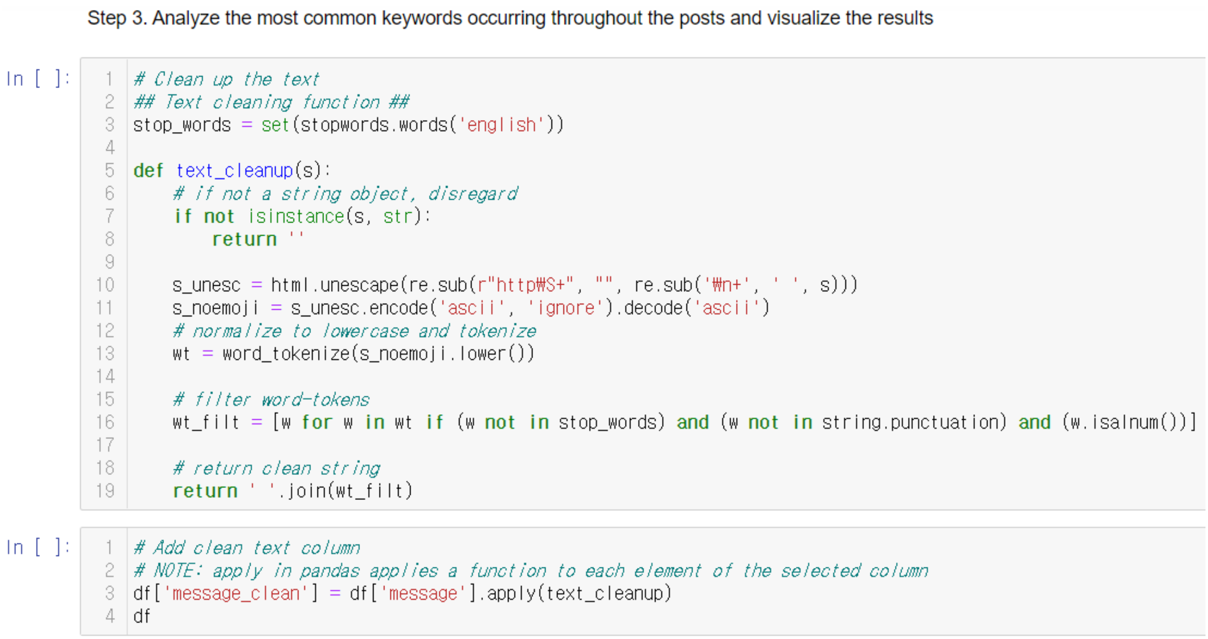
Follow the given analysis steps :

Step 1. Identify what types of posts in the dataset and how many entries of each post type are present

In [ ]:
```
# Frequency for each type
for tp in df["type"].unique():
    ntp = len(df[df["type"] == tp])
    print(f"Type {tp} occurs {ntp} times")
```

Step 2. List the average number of shares for each post type

In [ ]:
```
# Average of each type of shares
for etp in df["type"].unique():
    share = df[df["type"] == etp]["shares_count"].mean()
    print(f"Type {etp} shared {share.round(decimals = 2)} times on average")
```



Step 3. Analyze the most common keywords occurring throughout the posts and visualize the results

In [ ]:
```
# Clean up the text
## Text cleaning function ##
stop_words = set(stopwords.words('english'))

def text_cleanup(s):
    # if not a string object, disregard
    if not isinstance(s, str):
        return ''

    s_unesc = html.unescape(re.sub(r"http₩S+", "", re.sub('₩n+', ' ', s)))
    s_noemoji = s_unesc.encode('ascii', 'ignore').decode('ascii')
    # normalize to lowercase and tokenize
    wt = word_tokenize(s_noemoji.lower())

    # filter word-tokens
    wt_filt = [w for w in wt if (w not in stop_words) and (w not in string.punctuation) and (w.isalnum())]

    # return clean string
    return ' '.join(wt_filt)
```

In [ ]:
```
# Add clean text column
# NOTE: apply in pandas applies a function to each element of the selected column
df['message_clean'] = df['message'].apply(text_cleanup)
df
```

In [ ]:

```python
# Combine all post text entries
text_all = ' '.join(df['message_clean'])

# Keywords occurring throughout all of the posts
wc = WordCloud(width=1200, height=800, max_font_size=110, collocations=False).generate(text_all)
plt.axis("off")
plt.imshow(wc, interpolation="bilinear")
plt.show()
```

In [ ]:

```python
# Store the words used to create WordCloud as kwords
kwords = WordCloud().process_text(text_all)

# Transform that dictionary into a pandas DataFrame
df_kwords = pd.DataFrame(list(kwords.items()), columns=['keyword', 'count']).set_index('keyword')

# Plot a bar chart with the top keywords
%matplotlib inline
df_kwords.sort_values(by='count', ascending=False).head(20).plot.bar()
```

Step 4. Explore the times when posts were created for the most shared entries

In [ ]:

```python
df.sort_values(by='shares_count', ascending=False).head(20)['created_time']
```

Step 5. Identify the top-20 posts that received the most shares

In [ ]:

```python
# Top-shared entries: combined text
text_shared = ' '.join(df.sort_values(by='shares_count', ascending=False)['message_clean'].head(20))

# Proceed generating a word cloud
wc = WordCloud(width=1200, height=800, max_font_size=110, collocations=False).generate(text_shared)
plt.axis("off")
plt.imshow(wc, interpolation="bilinear")
plt.show()
```

In [ ]:

```python
# Store the words used to create WordCloud as kwords1
kwords1 = WordCloud().process_text(text_shared)

# Transform that dictionary into a pandas DataFrame
df_kwords1 = pd.DataFrame(list(kwords1.items()), columns=['keyword', 'count']).set_index('keyword')

# Plot a bar chart with the top keywords
%matplotlib inline
df_kwords1.sort_values(by='count', ascending=False).head(20).plot.bar()
```

Additional Step. Analyze the relationship between each count

In [ ]:

```
# Convert correlation matrix into a DataFrame
df_corr = df.corr(method = 'pearson').round(decimals = 2)

# specifying plot size (making it bigger)
fig, ax = plt.subplots(figsize=(12,12))


# developing a spicy heatmap
sns.heatmap(data       = df_corr, # the correlation matrix
            cmap       = 'inferno',      # changing to SPICY colors
            square     = True,          # tightening the layout
            annot      = True,          # should there be numbers in the heatmap
            linecolor  = 'black',       # lines between boxes
            linewidths = 0.5)           # how thick should the lines be?


# title and displaying the plot
plt.title("""
Linear Correlation Heatmap for each count
""")

plt.show()
```