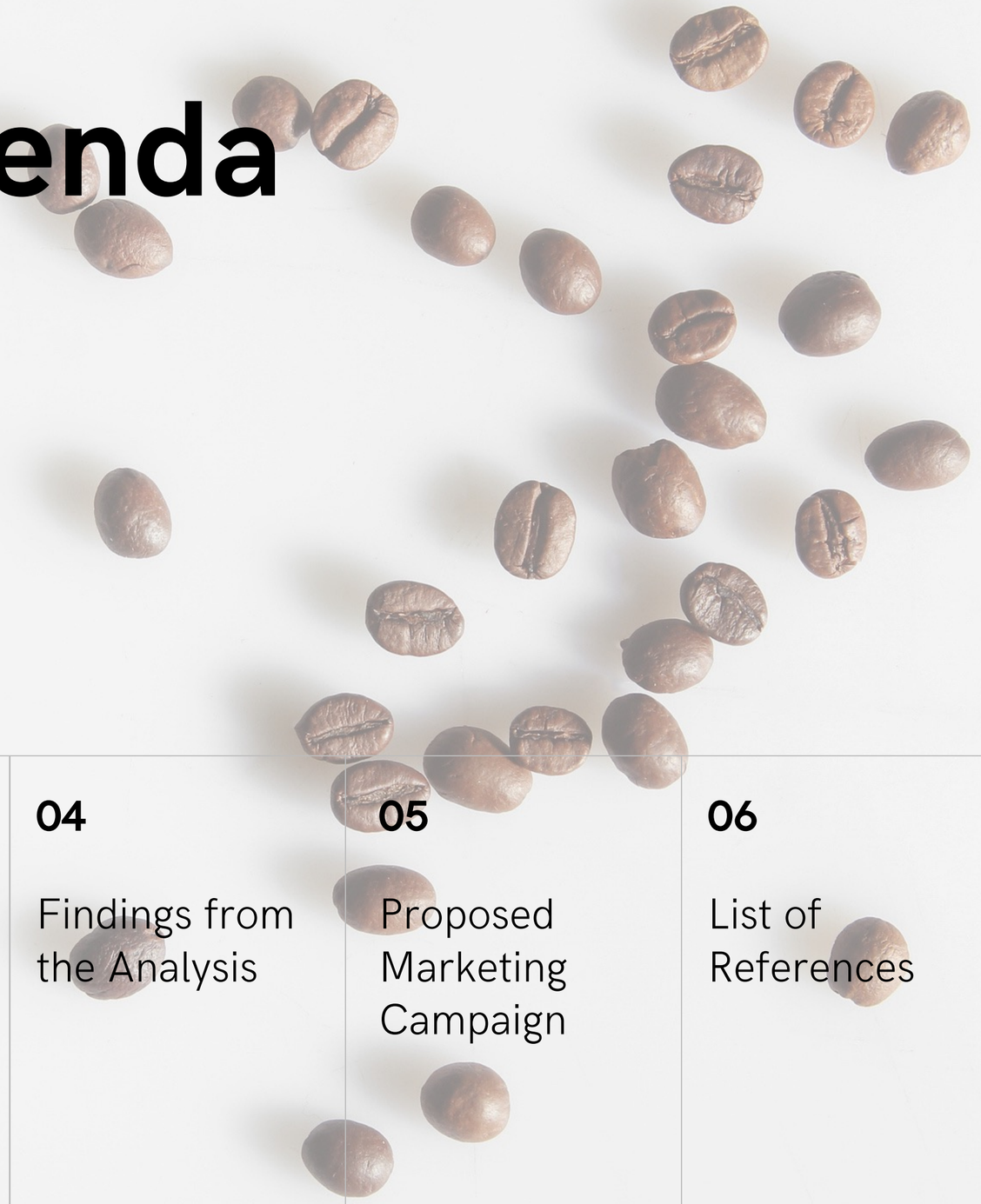
A1 **Social Media Analytics Project** | Social Media Analytics for Business Intelligence | Martine Elisabeth Stø Johansen

Coffee Brand Strategy

Presentation Agenda

Key topics from analysis



01

Executive Summary and Highlights

02

Highlights from Analysis

03

Competitor Analysis

04

Findings from the Analysis

05

Proposed Marketing Campaign

06

List of References

Executive Summary

New Coffee Capsule or Pod Brand Social Media Strategy

As an analyst for a business intelligence firm, I have analyzed the coffee capsule and pod market to guide a new startup to understand what to focus on and to take better choices for their social media marketing plan.

The analysis is based on tweets from about major players in the industry, as well as key features of each brand. The purpose is to figure out what system to create for, and what type of features costumers prefer. The focus is on the coffee, rather than the machines. Not looking at financial factors, but market perception.

New trends in the market is an increasing focus on environmental factors, but does this affect the sentiment on the brands?

Highlights from Analysis

A brief overview of key findings from my analysis of the coffee capsule and pod market.

The Most Positively Viewed System: Vertuo

Based on brand sentiments and feature sentiments, Nespresso's new system "Vertuo" is the coffee capsule system with the most positive sentiments.

Foam is More Attractive than *Crema*

Nespresso is know for it´s crema, but analysis shows that foam is (surprisingly) more positively tweeted about than crema.

Customer Service

Most retweeted tweet from Keurig is related to great customer service. That is valuable insight about what engage customers.

The Global Coffee Capsule and Pod Market

Expected Growth Rate Through 2023

$5.2 Billion

Expected Growth Rate Trough 2023

6.77%

Convenient and instant prepare single servings coffee are seeing a growing popularity. Coffee capsules contains fine ground coffee that through several different systems provide chosen amount and flavor of fresh coffee. There are different machines that works with different capsules systems, so new brands have to decide what system to go for. The most well known systems are Nespresso, Keurig, Nespresso's Vertuo, Nescafé and Lavazza. They differ not only in shape of capsules/pods, but in choice of amount coffee, flavors, foam and crema. A newer trend in the market is an increasingly focus on environment, that several brands have done adjustments after.

Source: The Business Research Company (2021).

Brand Sentiments

The following contains a brand sentiment analysis of major players in the industry of coffee capsule and pod market. Looking into sentiments on key features: price/cost, crema, foam, taste/flavor, roast, size and environment.

NESPRESSO

0.131902 1

KEURIG

0.176279 2

VERTUO

0.252976 3

NESCAFE

-0.008333 4

LAVAZZA

-0.017857

NESPRESSO

Breand Feature Sentiment Analysis

High Sentiment on Flavor/Taste

Nespresso's highest sentiment is on favor and taste, as they have a large range of alternatives. Size is the lowest, where they provide 3 different sizes

Feature Sentiments

Features	*Average Sentiment*
Price	*0.091927*
Cost	*0.080612*
Crema	*0.000000*
Foam	*NaN*
Taste	*0.239063*
Flavor	*0.283333*
Roast	*0.180124*
Size	*-0.160000*
Environment	*0.220357*

KEURIG®

Breand Feature Sentiment Analysis

High Sentiment on Foam and Taste

Highest sentiment on foam and taste.
Negative sentiment on cost and size

Feature Sentiments

Features	*Average Sentiment*
Price	0.026305
Cost	0.102083
Crema	NaN
Foam	0.375000
Taste	0.262108
Flavor	NaN
Roast	0.195799
Size	-0.075649
Environment	-0.152500

NESPRESSO®

VERTUO

Breand Feature Sentiment Analysis

Overall Highest Sentiment on Foam

Highest overall score on foam, and surprisingly positive sentiment on price and cost. High sentiment on taste too. Size is neutral, that was one of Nespresso's main focuses when launching Vertuo.

Feature Sentiments

Features	*Average Sentiment*
Price	0.320152
Cost	0.250000
Crema	NaN
Foam	0.550000
Taste	0.230556
Flavor	NaN
Roast	0.087500
Size	0.000000
Environment	NaN

NESCAFÉ

Breand Feature Sentiment Analysis

High Sentiment on Environment

Overall highest sentiment on environment.
Negative sentiment on foam and size.

Feature Sentiments

Features	*Average Sentiment*
Price	0.011356
Cost	0.210665
Crema	NaN
Foam	-0.066667
Taste	0.235934
Flavor	0.062731
Roast	0.103958
Size	-0.008333
Environment	0.484091

LAVAZZA

Breand Feature Sentiment Analysis

High Sentiment on Price/Cost

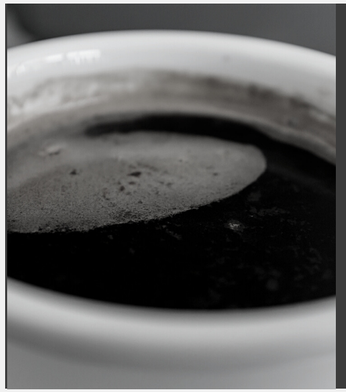
High score on price and cost, witch is their focus. Few tweets, but the few are positively regarding features.

Feature Sentiments

Features	*Average Sentiment*
Price	0.455556
Cost	0.316667
Crema	0.198016
Foam	NaN
Taste	0.366667
Flavor	NaN
Roast	0.216250
Size	NaN
Environment	0.400000

Findings from the Analysis

Based on the analysis, here are what new entries should focus on, systems to choose and marketing strategies.



01 Focus

Features of most importance for customers is shown to be foam, flavor and price. Environmental factors may be a trend, but the tweets did not provide any data here.

03 Social Media Marketing

The word “price” is viewed as more positive than “cost”. Essential to choose word wisely while creating content. Focus on what people want: foam, large rang of flavors and good prices!

02 Producers

The system that shown to be most attractive is Vertuo capsule systems. Vertuo differ with choice of larger cup size, more foam and great value. However, it´s not possible to create for them due to patent. Therefore, it´s recommended to create for Keurig.

Marketing inputs

From Keurig analysis, the recommended system

Most Retweeted Example

'Buying this Keurig was the best thing ever, Ive had 3 cappuccinos already 😂😂'

Most Retweeted Example

'yo yo the keurig lady on the phone today, i hope ur having a wonderful day and ily thank u sm for helping me T-T'

Popular keywords

Gift, Office, Work, Brewer, Hot, New, People, Today, Cup.

Key Influencers

Trumpeteer14
Froschauer_AF
KelliGOPHATER
CalTeamstersPAC
VVDailyPress
mattalley413
JakeThompsonOE
steelworkers
PeteGGriffin
HarborMom
mypenplease
Keurig
HandbagsByDyana
Herbworthy
fuzweb
wbm312
mypenplease
GingaHerb
mypenplease
LombardiHimself

Social Media Marketing Campaign

01

New Product Focus

Creating pods for the Keurig system, focusing on replicating the Vertuo profile, with great foam and taste, focusing on real coffee flavors, and as always having great customer service.

02

Promotion

Promote to the right people, coffee influencers exists, and they are most often coffee purists. Advocating the true coffee flavors to them can help them spread the new products.

03

Example Marketing Tweet

Based on the word cloud and content in most retweeted tweets, an example marketing tweet could be:

"The perfect gift, brewing a better start to the work day"

Thank you for your time!

Please feel free to send me any questions you may have.

LIST OF REFERENCES

The Business Research Company. (2021). *Coffee Capsule Global Market Report 2021: COVID-19 Growth And Change To 2030*. Retrieved from: https://www.thebusinessresearchcompany.com/report/coffee-capsule-market-global-report-2020-30-covid-19-growth-and-change

Twitter. (2021). Use Cases, Tutorials, & Documentation | Twitter Developer Platform. Twitter. https://developer.twitter.com/en.

A1: Social Media Analytics Project

Martine Elisabeth Stø Johansen - Student ID: 10874334

In [1]:

```
import json
import tweepy as tw
```

In [2]:

```
import config_twitter
```

In [3]:

```
# function to establish an initial API connection, respecting the rate lim
def connect_api_client():
    auth = tw.OAuthHandler(config_twitter.consumer_key, config_twitter.cons
    auth.set_access_token(config_twitter.access_token, config_twitter.acces
    api = tw.API(auth, wait_on_rate_limit=True)
    try:
        # returns False if credentials could not be verified
        api.verify_credentials()
        user = api.verify_credentials()
        if not user:
            raise("Credentials could not be verified: Please check config.p
        print(f"Connected to Twitter API as {user.name}")
    except Exception as e:
        raise e
    return api

api = connect_api_client()

# constructing a search query
query = '"coffee capsules" OR "coffee capsule" OR "coffee pods" OR "coffee

# decideing how many tweets to query
ntweets = 2000

# search and collect relevant tweets
tweets = [tweet._json for tweet in tw.Cursor(api.search, q=query, lang="en

# save tweets data to json file
file_out = f"raw_tweet_data_coffee_capsules_{ntweets}.json"
with open(file_out, mode='w') as f:
    f.write(json.dumps(tweets, indent=2))
```

```
Connected to Twitter API as Martine Elisabeth Stø Johansen
```

In [4]:

```
# check number of tweets in dataset
len(tweets)
```

Out[4]:

```
398
```

Few tweets in dataset, reduction from previous runs. Will collect from each brand to receive more data

In [5]:

```
# checking a random tweet to controll content
tweets[13]
```

Out[5]:


```
{'created_at': 'Fri Jul 30 16:12:30 +0000 2021',
 'id': 1421141662802223111,
 'id_str': '1421141662802223111',
 'full_text': 'illy coffee capsule machine😭 https://t.co/qTIOsSjLYP',
 'truncated': False,
 'display_text_range': [0, 28],
 'entities': {'hashtags': [],
  'symbols': [],
  'user_mentions': [],
  'urls': [{'url': 'https://t.co/qTIOsSjLYP',
    'expanded_url': 'https://twitter.com/etherealdior/status/14211083173219
81957',
    'display_url': 'twitter.com/etherealdior/s…',
    'indices': [29, 52]}]},
 'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
 'source': '<a href="http://twitter.com/download/iphone" rel="nofollow">Twi
tter for iPhone</a>',
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'in_reply_to_screen_name': None,
 'user': {'id': 1010947550805803009,
  'id_str': '1010947550805803009',
  'name': '😊',
  'screen_name': 'amengmeongmeng',
  'location': 'Singapore ',
  'description': 'harry styles, food, baking cakes, skincare nerd, clean-fr
eak, overly scented laundry, neat kitchen, clean rooms, essential oils and
diffuser avid collector😍',
  'url': 'https://t.co/DjaYIwkZvu',
  'entities': {'url': {'urls': [{'url': 'https://t.co/DjaYIwkZvu',
      'expanded_url': 'https://link.medium.com/NdXdX3pUl5',
      'display_url': 'link.medium.com/NdXdX3pUl5',
      'indices': [0, 23]}]},
   'description': {'urls': []}},
  'protected': False,
  'followers_count': 102,
  'friends_count': 237,
  'listed_count': 0,
  'created_at': 'Sun Jun 24 18:07:33 +0000 2018',
  'favourites_count': 4076,
  'utc_offset': None,
  'time_zone': None,
  'geo_enabled': True,
  'verified': False,
  'statuses_count': 10643,
  'lang': None,
  'contributors_enabled': False,
  'is_translator': False,
  'is_translation_enabled': False,
```

```
  'profile_background_color': 'F5F8FA',
  'profile_background_image_url': None,
  'profile_background_image_url_https': None,
  'profile_background_tile': False,
  'profile_image_url': 'http://pbs.twimg.com/profile_images/140664932660920
3201/71ZN6W0c_normal.jpg',
  'profile_image_url_https': 'https://pbs.twimg.com/profile_images/14066493
26609203201/71ZN6W0c_normal.jpg',
  'profile_banner_url': 'https://pbs.twimg.com/profile_banners/101094755080
5803009/1621671628',
  'profile_link_color': '1DA1F2',
  'profile_sidebar_border_color': 'C0DEED',
  'profile_sidebar_fill_color': 'DDEEF6',
  'profile_text_color': '333333',
  'profile_use_background_image': True,
  'has_extended_profile': True,
  'default_profile': True,
  'default_profile_image': False,
  'following': False,
  'follow_request_sent': False,
  'notifications': False,
  'translator_type': 'none',
  'withheld_in_countries': []},
 'geo': None,
 'coordinates': None,
 'place': None,
 'contributors': None,
 'is_quote_status': True,
 'quoted_status_id': 1421108317321981957,
 'quoted_status_id_str': '1421108317321981957',
 'quoted_status': {'created_at': 'Fri Jul 30 14:00:00 +0000 2021',
  'id': 1421108317321981957,
  'id_str': '1421108317321981957',
  'full_text': 'how i want my life to look https://t.co/21gC9xVXpH',
  'truncated': False,
  'display_text_range': [0, 26],
  'entities': {'hashtags': [],
   'symbols': [],
   'user_mentions': [],
   'urls': [],
   'media': [{'id': 1420151164528734209,
     'id_str': '1420151164528734209',
     'indices': [27, 50],
     'media_url': 'http://pbs.twimg.com/media/E7Vj9bkXsAEu-cA.png',
     'media_url_https': 'https://pbs.twimg.com/media/E7Vj9bkXsAEu-cA.png',
     'url': 'https://t.co/21gC9xVXpH',
     'display_url': 'pic.twitter.com/21gC9xVXpH',
     'expanded_url': 'https://twitter.com/etherealdior/status/1421108317321
981957/photo/1',
     'type': 'photo',
     'sizes': {'medium': {'w': 500, 'h': 494, 'resize': 'fit'},
      'small': {'w': 500, 'h': 494, 'resize': 'fit'},
      'large': {'w': 500, 'h': 494, 'resize': 'fit'},
      'thumb': {'w': 150, 'h': 150, 'resize': 'crop'}}}]},
  'extended_entities': {'media': [{'id': 1420151164528734209,
     'id_str': '1420151164528734209',
     'indices': [27, 50],
     'media_url': 'http://pbs.twimg.com/media/E7Vj9bkXsAEu-cA.png',
     'media_url_https': 'https://pbs.twimg.com/media/E7Vj9bkXsAEu-cA.png',
     'url': 'https://t.co/21gC9xVXpH',
     'display_url': 'pic.twitter.com/21gC9xVXpH',
```

```
     'expanded_url': 'https://twitter.com/etherealdior/status/1421108317321
981957/photo/1',
     'type': 'photo',
     'sizes': {'medium': {'w': 500, 'h': 494, 'resize': 'fit'},
      'small': {'w': 500, 'h': 494, 'resize': 'fit'},
      'large': {'w': 500, 'h': 494, 'resize': 'fit'},
      'thumb': {'w': 150, 'h': 150, 'resize': 'crop'}}},
    {'id': 1420151164541276165,
     'id_str': '1420151164541276165',
     'indices': [27, 50],
     'media_url': 'http://pbs.twimg.com/media/E7Vj9bnXEAU7i4j.png',
     'media_url_https': 'https://pbs.twimg.com/media/E7Vj9bnXEAU7i4j.png',
     'url': 'https://t.co/21gC9xVXpH',
     'display_url': 'pic.twitter.com/21gC9xVXpH',
     'expanded_url': 'https://twitter.com/etherealdior/status/1421108317321
981957/photo/1',
     'type': 'photo',
     'sizes': {'large': {'w': 564, 'h': 564, 'resize': 'fit'},
      'medium': {'w': 564, 'h': 564, 'resize': 'fit'},
      'small': {'w': 564, 'h': 564, 'resize': 'fit'},
      'thumb': {'w': 150, 'h': 150, 'resize': 'crop'}}},
    {'id': 1420151164528693249,
     'id_str': '1420151164528693249',
     'indices': [27, 50],
     'media_url': 'http://pbs.twimg.com/media/E7Vj9bkXEAEzAGf.png',
     'media_url_https': 'https://pbs.twimg.com/media/E7Vj9bkXEAEzAGf.png',
     'url': 'https://t.co/21gC9xVXpH',
     'display_url': 'pic.twitter.com/21gC9xVXpH',
     'expanded_url': 'https://twitter.com/etherealdior/status/1421108317321
981957/photo/1',
     'type': 'photo',
     'sizes': {'small': {'w': 554, 'h': 554, 'resize': 'fit'},
      'large': {'w': 554, 'h': 554, 'resize': 'fit'},
      'medium': {'w': 554, 'h': 554, 'resize': 'fit'},
      'thumb': {'w': 150, 'h': 150, 'resize': 'crop'}}},
    {'id': 1420151164528730115,
     'id_str': '1420151164528730115',
     'indices': [27, 50],
     'media_url': 'http://pbs.twimg.com/media/E7Vj9bkXoAMrxZ4.png',
     'media_url_https': 'https://pbs.twimg.com/media/E7Vj9bkXoAMrxZ4.png',
     'url': 'https://t.co/21gC9xVXpH',
     'display_url': 'pic.twitter.com/21gC9xVXpH',
     'expanded_url': 'https://twitter.com/etherealdior/status/1421108317321
981957/photo/1',
     'type': 'photo',
     'sizes': {'medium': {'w': 564, 'h': 564, 'resize': 'fit'},
      'small': {'w': 564, 'h': 564, 'resize': 'fit'},
      'thumb': {'w': 150, 'h': 150, 'resize': 'crop'},
      'large': {'w': 564, 'h': 564, 'resize': 'fit'}}}]},
  'metadata': {'iso_language_code': 'en', 'result_type': 'recent'},
  'source': '<a href="https://mobile.twitter.com" rel="nofollow">Twitter We
b App</a>',
  'in_reply_to_status_id': None,
  'in_reply_to_status_id_str': None,
  'in_reply_to_user_id': None,
  'in_reply_to_user_id_str': None,
  'in_reply_to_screen_name': None,
  'user': {'id': 941544401662181376,
   'id_str': '941544401662181376',
   'name': 'stella 🕊️',
```

```
    'screen_name': 'etherealdior',
    'location': 'moodboard ig: etherealmuses',
    'description': "we'll always have paris | photos are not mine unless oth
erwise stated | she/her | https://t.co/DkSU1ZEz2P",
    'url': 'https://t.co/2kK14Z0YNn',
    'entities': {'url': {'urls': [{'url': 'https://t.co/2kK14Z0YNn',
        'expanded_url': 'https://curiouscat.me/ethereaIdior',
        'display_url': 'curiouscat.me/ethereaIdior',
        'indices': [0, 23]}]},
     'description': {'urls': [{'url': 'https://t.co/DkSU1ZEz2P',
        'expanded_url': 'http://pinterest.ca/ethereaIdior/',
        'display_url': 'pinterest.ca/ethereaIdior/',
        'indices': [82, 105]}]}},
    'protected': False,
    'followers_count': 50991,
    'friends_count': 44,
    'listed_count': 350,
    'created_at': 'Fri Dec 15 05:44:13 +0000 2017',
    'favourites_count': 7669,
    'utc_offset': None,
    'time_zone': None,
    'geo_enabled': False,
    'verified': False,
    'statuses_count': 6644,
    'lang': None,
    'contributors_enabled': False,
    'is_translator': False,
    'is_translation_enabled': False,
    'profile_background_color': 'F5F8FA',
    'profile_background_image_url': None,
    'profile_background_image_url_https': None,
    'profile_background_tile': False,
    'profile_image_url': 'http://pbs.twimg.com/profile_images/13689258132199
91552/OjGLPmq__normal.jpg',
    'profile_image_url_https': 'https://pbs.twimg.com/profile_images/1368925
813219991552/OjGLPmq__normal.jpg',
    'profile_banner_url': 'https://pbs.twimg.com/profile_banners/94154440166
2181376/1615212322',
    'profile_link_color': '1DA1F2',
    'profile_sidebar_border_color': 'C0DEED',
    'profile_sidebar_fill_color': 'DDEEF6',
    'profile_text_color': '333333',
    'profile_use_background_image': True,
    'has_extended_profile': True,
    'default_profile': True,
    'default_profile_image': False,
    'following': False,
    'follow_request_sent': False,
    'notifications': False,
    'translator_type': 'none',
    'withheld_in_countries': []},
   'geo': None,
   'coordinates': None,
   'place': None,
   'contributors': None,
   'is_quote_status': False,
   'retweet_count': 346,
   'favorite_count': 1365,
   'favorited': False,
   'retweeted': False,
   'possibly_sensitive': False,
```

```
 'lang': 'en'},
'retweet_count': 0,
'favorite_count': 1,
'favorited': False,
'retweeted': False,
'possibly_sensitive': False,
'lang': 'en'}
```

Will run several to get more data

In [6]:

```
# Twitter data analysis starter kit
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 pandas as pd
import matplotlib.pyplot as plt
```

In [7]:

```
fjson = 'raw_tweet_data_coffee_capsules_2000.json'

with open(fjson) as file:
    data = json.load(file)

df_tweets = pd.DataFrame([t['full_text'] for t in data], columns=['text'])

df_tweets['retweets'] = [t['retweet_count'] for t in data]
df_tweets['favorites'] = [t['favorite_count'] for t in data]
df_tweets['user'] = [t['user']['screen_name'] for t in data]

stop_words = set(stopwords.words('english'))

def text_cleanup(s):
    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
    # return clean string
    return ' '.join(wt_filt)

df_tweets['text_clean'] = df_tweets['text'].apply(text_cleanup)

def sentim_polarity(s):
    return TextBlob(s).sentiment.polarity

def sentim_subject(s):
    return TextBlob(s).sentiment.subjectivity

df_tweets['polarity'] = df_tweets['text_clean'].apply(sentim_polarity)
df_tweets['subjectivity'] = df_tweets['text_clean'].apply(sentim_subject)
```

In [8]:

```
brands = ['nespresso', 'keurig', 'vertuo', 'nescafe', 'lavazza']

df_brands = pd.DataFrame(brands, columns=['brand'])

def brand_sentiment(b):
    return df_tweets[df_tweets['text_clean'].str.contains(b)]['polarity'].r

df_brands['average_sentiment'] = df_brands['brand'].apply(brand_sentiment)

print("competitive brands")
print(df_brands)
```

```
competitive brands
       brand  average_sentiment
0  nespresso           0.131902
1     keurig           0.176279
2     vertuo           0.252976
3    nescafe          -0.008333
4    lavazza          -0.017857
```

In [30]:

```
# construct a search query for Nespresso
query = '"nespresso" -filter:retweets'

ntweets = 2000

tweets_nespresso = [tweet._json for tweet in tw.Cursor(api.search, q=query,

file_out = f"raw_tweet_data_nespresso_{ntweets}.json"
with open(file_out, mode='w') as f:
    f.write(json.dumps(tweets, indent=2))
```

In [31]:

```
len(tweets_nespresso)
```

Out[31]:

```
1298
```

In [32]:

```
fjson_nespresso = 'raw_tweet_data_nespresso_2000.json'

with open(fjson_nespresso) as file:
    data = json.load(file)

df_tweets_nespresso = pd.DataFrame([t['full_text'] for t in data], columns=

df_tweets_nespresso['retweets'] = [t['retweet_count'] for t in data]
df_tweets_nespresso['favorites'] = [t['favorite_count'] for t in data]
df_tweets_nespresso['user'] = [t['user']['screen_name'] for t in data]

stop_words = set(stopwords.words('english'))

def text_cleanup(s):
    s_unesc = html.unescape(re.sub(r"http\S+", "", re.sub('\n+', ' ', s)))
    s_noemoji = s_unesc.encode('ascii', 'ignore').decode('ascii')
    wt = word_tokenize(s_noemoji.lower())

    wt_filt = [w for w in wt if (w not in stop_words) and (w not in string

    return ' '.join(wt_filt)

df_tweets_nespresso['text_clean'] = df_tweets_nespresso['text'].apply(text_

def sentim_polarity(s):
    return TextBlob(s).sentiment.polarity

def sentim_subject(s):
    return TextBlob(s).sentiment.subjectivity

df_tweets_nespresso['polarity'] = df_tweets_nespresso['text_clean'].apply(s
df_tweets_nespresso['subjectivity'] = df_tweets_nespresso['text_clean'].ap

features = ['price', 'cost', 'crema', 'foam', 'taste', 'flavour', 'roast',

df_features_nespresso = pd.DataFrame(features, columns=['features'])

def feature_sentiment(b):
    return df_tweets_nespresso[df_tweets_nespresso['text_clean'].str.conta

df_features_nespresso['average_sentiment'] = df_features_nespresso['feature

print("feature")
print(df_features_nespresso)
```

```
feature
      features  average_sentiment
0        price           0.091927
1         cost           0.080612
2        crema           0.000000
3         foam                NaN
4        taste           0.239063
5      flavour           0.283333
6        roast           0.180124
7         size          -0.160000
8  environment           0.220357
```

In [12]:

```
# construct a search query Keurig
query = '"keurig" -filter:retweets'

ntweets = 2000

tweets_keurig = [tweet._json for tweet in tw.Cursor(api.search, q=query, la
len(tweets_keurig)

file_out = f"raw_tweet_data_keurig_{ntweets}.json"
with open(file_out, mode='w') as f:
    f.write(json.dumps(tweets_keurig, indent=2))
```

In [13]:

```
len(tweets_keurig)
```

Out[13]: 1777

In [14]:

```
fjson_keurig = 'raw_tweet_data_keurig_2000.json'

with open(fjson_keurig) as file:
    data = json.load(file)

df_tweets_keurig = pd.DataFrame([t['full_text'] for t in data], columns=['t

df_tweets_keurig['retweets'] = [t['retweet_count'] for t in data]
df_tweets_keurig['favorites'] = [t['favorite_count'] for t in data]
df_tweets_keurig['user'] = [t['user']['screen_name'] for t in data]

stop_words = set(stopwords.words('english'))

def text_cleanup(s):
    s_unesc = html.unescape(re.sub(r"http\S+", "", re.sub('\n+', ' ', s)))
    s_noemoji = s_unesc.encode('ascii', 'ignore').decode('ascii')
    wt = word_tokenize(s_noemoji.lower())

    wt_filt = [w for w in wt if (w not in stop_words) and (w not in string

    return ' '.join(wt_filt)

df_tweets_keurig['text_clean'] = df_tweets_keurig['text'].apply(text_cleanu

def sentim_polarity(s):
    return TextBlob(s).sentiment.polarity

def sentim_subject(s):
    return TextBlob(s).sentiment.subjectivity

df_tweets_keurig['polarity'] = df_tweets_keurig['text_clean'].apply(sentim_
df_tweets_keurig['subjectivity'] = df_tweets_keurig['text_clean'].apply(sen

features = ['price', 'cost', 'crema', 'foam', 'taste', 'flavour', 'roast',

df_features_keurig = pd.DataFrame(features, columns=['features'])


def feature_sentiment(b):
    return df_tweets_keurig[df_tweets_keurig['text_clean'].str.contains(b)

df_features_keurig['average_sentiment'] = df_features_keurig['features'].ap

print("feature")
print(df_features_keurig)
```

```
feature
      features  average_sentiment
0        price           0.026305
1         cost           0.102083
2        crema                NaN
3         foam           0.375000
4        taste           0.262108
5      flavour                NaN
6        roast           0.195799
7         size          -0.075649
8  environment          -0.152500
```

In [15]:

```
# construct a search query Vertuo
query = '"vertuo" -filter:retweets'

ntweets = 2000

tweets_vertuo = [tweet._json for tweet in tw.Cursor(api.search, q=query, la
len(tweets_vertuo)

file_out = f"raw_tweet_data_vertuo_{ntweets}.json"
with open(file_out, mode='w') as f:
    f.write(json.dumps(tweets_vertuo, indent=2))
```

In [16]:

```
len(tweets_vertuo)
```

Out[16]:

```
81
```

In [17]:

```
fjson_vertuo = 'raw_tweet_data_vertuo_2000.json'

with open(fjson_vertuo) as file:
    data = json.load(file)

df_tweets_vertuo = pd.DataFrame([t['full_text'] for t in data], columns=['t

df_tweets_vertuo['retweets'] = [t['retweet_count'] for t in data]
df_tweets_vertuo['favorites'] = [t['favorite_count'] for t in data]
df_tweets_vertuo['user'] = [t['user']['screen_name'] for t in data]

stop_words = set(stopwords.words('english'))

def text_cleanup(s):
    s_unesc = html.unescape(re.sub(r"http\S+", "", re.sub('\n+', ' ', s)))
    s_noemoji = s_unesc.encode('ascii', 'ignore').decode('ascii')
    wt = word_tokenize(s_noemoji.lower())

    wt_filt = [w for w in wt if (w not in stop_words) and (w not in string

    return ' '.join(wt_filt)

df_tweets_vertuo['text_clean'] = df_tweets_vertuo['text'].apply(text_cleanu

def sentim_polarity(s):
    return TextBlob(s).sentiment.polarity

def sentim_subject(s):
    return TextBlob(s).sentiment.subjectivity

df_tweets_vertuo['polarity'] = df_tweets_vertuo['text_clean'].apply(sentim_
df_tweets_vertuo['subjectivity'] = df_tweets_vertuo['text_clean'].apply(se

features = ['price', 'cost', 'crema', 'foam', 'taste', 'flavour', 'roast',

df_features_vertuo = pd.DataFrame(features, columns=['features'])


def feature_sentiment(b):
    return df_tweets_vertuo[df_tweets_vertuo['text_clean'].str.contains(b)

df_features_vertuo['average_sentiment'] = df_features_vertuo['features'].a

print("feature")
print(df_features_vertuo)
```

```
feature
      features  average_sentiment
0        price           0.320152
1         cost           0.250000
2        crema                NaN
3         foam           0.550000
4        taste           0.230556
5      flavour                NaN
6        roast           0.087500
7         size           0.000000
8  environment                NaN
```

In [18]:

```
# construct a search query Nescafe
query = '"Nescafe" -filter:retweets'

ntweets = 2000

tweets_Nescafe = [tweet._json for tweet in tw.Cursor(api.search, q=query, 
len(tweets_Nescafe)

file_out = f"raw_tweet_data_Nescafe_{ntweets}.json"
with open(file_out, mode='w') as f:
    f.write(json.dumps(tweets_Nescafe, indent=2))
```

In [19]:

```
len(tweets_Nescafe)
```

Out[19]:

```
1598
```

In [20]:

```
fjson_Nescafe = 'raw_tweet_data_Nescafe_2000.json'

with open(fjson_Nescafe) as file:
    data = json.load(file)

df_tweets_Nescafe = pd.DataFrame([t['full_text'] for t in data], columns=[

df_tweets_Nescafe['retweets'] = [t['retweet_count'] for t in data]
df_tweets_Nescafe['favorites'] = [t['favorite_count'] for t in data]
df_tweets_Nescafe['user'] = [t['user']['screen_name'] for t in data]

stop_words = set(stopwords.words('english'))

def text_cleanup(s):
    s_unesc = html.unescape(re.sub(r"http\S+", "", re.sub('\n+', ' ', s)))
    s_noemoji = s_unesc.encode('ascii', 'ignore').decode('ascii')
    wt = word_tokenize(s_noemoji.lower())

    wt_filt = [w for w in wt if (w not in stop_words) and (w not in string

    return ' '.join(wt_filt)

df_tweets_Nescafe['text_clean'] = df_tweets_Nescafe['text'].apply(text_clea

def sentim_polarity(s):
    return TextBlob(s).sentiment.polarity

def sentim_subject(s):
    return TextBlob(s).sentiment.subjectivity

df_tweets_Nescafe['polarity'] = df_tweets_Nescafe['text_clean'].apply(senti
df_tweets_Nescafe['subjectivity'] = df_tweets_Nescafe['text_clean'].apply(s

features = ['price', 'cost', 'crema', 'foam', 'taste', 'flavour', 'roast',

df_features_Nescafe = pd.DataFrame(features, columns=['features'])


def feature_sentiment(b):
    return df_tweets_Nescafe[df_tweets_Nescafe['text_clean'].str.contains(b

df_features_Nescafe['average_sentiment'] = df_features_Nescafe['features']

print("feature")
print(df_features_Nescafe)
```

```
feature
      features  average_sentiment
0        price           0.011356
1         cost           0.210665
2        crema                NaN
3         foam          -0.066667
4        taste           0.235934
5      flavour           0.062731
6        roast           0.103958
7         size          -0.008333
8  environment           0.484091
```

In [22]:

```
# construct a search query Lavazza
query = '"Lavazza" -filter:retweets'

ntweets = 2000

tweets_Lavazza = [tweet._json for tweet in tw.Cursor(api.search, q=query, 
len(tweets_Lavazza)

file_out = f"raw_tweet_data_Lavazza_{ntweets}.json"
with open(file_out, mode='w') as f:
    f.write(json.dumps(tweets_Lavazza, indent=2))
```

In [23]:

```
len(tweets_Lavazza)
```

Out[23]:

```
192
```

In [24]:

```
fjson_Lavazza = 'raw_tweet_data_Lavazza_2000.json'

with open(fjson_Lavazza) as file:
    data = json.load(file)

df_tweets_Lavazza = pd.DataFrame([t['full_text'] for t in data], columns=[

df_tweets_Lavazza['retweets'] = [t['retweet_count'] for t in data]
df_tweets_Lavazza['favorites'] = [t['favorite_count'] for t in data]
df_tweets_Lavazza['user'] = [t['user']['screen_name'] for t in data]

stop_words = set(stopwords.words('english'))

def text_cleanup(s):
    s_unesc = html.unescape(re.sub(r"http\S+", "", re.sub('\n+', ' ', s)))
    s_noemoji = s_unesc.encode('ascii', 'ignore').decode('ascii')
    wt = word_tokenize(s_noemoji.lower())

    wt_filt = [w for w in wt if (w not in stop_words) and (w not in string

    return ' '.join(wt_filt)

df_tweets_Lavazza['text_clean'] = df_tweets_Lavazza['text'].apply(text_clea

def sentim_polarity(s):
    return TextBlob(s).sentiment.polarity

def sentim_subject(s):
    return TextBlob(s).sentiment.subjectivity

df_tweets_Lavazza['polarity'] = df_tweets_Lavazza['text_clean'].apply(senti
df_tweets_Lavazza['subjectivity'] = df_tweets_Lavazza['text_clean'].apply(s

features = ['price', 'cost', 'crema', 'foam', 'taste', 'flavour', 'roast',

df_features_Lavazza = pd.DataFrame(features, columns=['features'])


def feature_sentiment(b):
    return df_tweets_Lavazza[df_tweets_Lavazza['text_clean'].str.contains(b

df_features_Lavazza['average_sentiment'] = df_features_Lavazza['features']

print("feature")
print(df_features_Lavazza)
```

```
feature
      features  average_sentiment
0        price           0.455556
1         cost           0.316667
2        crema           0.198016
3         foam                NaN
4        taste           0.366667
5      flavour                NaN
6        roast           0.216250
7         size                NaN
8  environment           0.400000
```

Based on this market analysis, the optimal choice would had been to create capsules that works well with Vertuo systems, but since only Nespesso can create this at the moment - the best choice is to create for Keurig, and adjust after what customers appreciate with Vertuo.

The following data will help set up a marketing campaign.

In [25]:

```
# Some examples of tweet content with the highest and the lowest sentiment
print("Highest:")
print(df_tweets_keurig.sort_values(by='polarity', ascending=False).head(10
print("")
print("Lowest:")
print(df_tweets_keurig.sort_values(by='polarity').head(10))
```

```
Highest:
                                                  text  retweets  favorite
s  \
1304  Buying this Keurig was the best thing ever, Iv...         0
1
983   Check out Bostons Best Coffee Pod &amp; Capsul...         0
0
244   @Jeanna350 @LAfromLINY @TheRealDee71 Father's ...         0
3
947   Check out Bostons Best Coffee Pod &amp; Capsul...         0
0
1248  Have you tried Boston's Best Coffee Pod &amp; ...         0
0
740   yo yo the keurig lady on the phone today, i ho...         0
2
161   Best Kcup Ever💯😁 https://t.co/FE9hErWcKN #coff...         0
0
760   Try Angelino's Gourmet Coffee and Tea! Perfect...         0
0
762   Keurig, electric pressure cookers l, and air f...         0
2
982   Check out Bostons Best Coffee Pod &amp; Capsul...         0
0

                 user                                        text_clean  \
1304      ValeeLaPena  buying keurig best thing ever ive 3 cappuccino...
983   CharliesTreasu1  check bostons best coffee pod capsule keurig v...
244   MikeRey61920141  jeanna350 lafromliny therealdee71 father day t...
947        simplyshay  check bostons best coffee pod capsule keurig v...
1248         bevsshop  tried boston best coffee pod capsule keurig 42...
740        yeezussush  yo yo keurig lady phone today hope ur wonderfu...
```

```
161       at_homewith  best kcup ever coffee coffeebreak coffeetime c...
760     SubacidicRose  try angelino gourmet coffee tea perfect keurig...
762        anitralove  keurig electric pressure cookers l air fryers ...
982   CharliesTreasu1  check bostons best coffee pod capsule keurig a...

      polarity  subjectivity
1304       1.0           0.3
983        1.0           0.3
244        1.0           0.3
947        1.0           0.3
1248       1.0           0.3
740        1.0           1.0
161        1.0           0.3
760        1.0           1.0
762        1.0           0.3
982        1.0           0.3

Lowest:
                                                   text  retweets  favorite
s  \
969   I want this to be real so bad, lmfao. Keurig c...         0
0
1594  Terrible day to be a Keurig. Cups of joe going...         0
5
1370  made breakfast, drank nasty ass coffee from a ...         1
6
612   @nahitsTeCoby Lmaooo at a manufacturing plant ...         0
1
607   I don't which of my friends was more disgusted...         0
0
95    i hate these coffee drinking hoes at my job be...         0
0
1154  @wallflowermeg I hate keurig coffee it’s so wa...         0
0
1022  @Japherh7 @JohnnySwift19 @JesseKellyDC Only yo...         0         1
1
1163  its not that i hate the keurig, its that i hat...         0
0
489   @mrbenwexler They hate Coke, MLB, US Olympics,...         0
0

                user                                        text_clean  \
969   bootleg_barista            want real bad lmfao keurig coming sass
1594  GeorgeHadfiel10  terrible day keurig cups joe going like milk d...
1370         hyvkwons  made breakfast drank nasty ass coffee keurig i...
612        fbvisualss  nahitstecoby lmaooo manufacturing plant rock h...
607   phillustrations  friends disgusted disappointed carnivore frien...
95     kylanbutterfly  hate coffee drinking hoes job every time fix t...
1154       loukamades           wallflowermeg hate keurig coffee watery
1022  dntfollowaldirt  japherh7 johnnyswift19 jessekellydc idiots tri...
1163   stfujanstevens                   hate keurig hate dont understand
489        PBRKessler  mrbenwexler hate coke mlb us olympics capital ...

      polarity  subjectivity
969     -1.000         1.000
1594    -1.000         1.000
1370    -1.000         1.000
612     -1.000         1.000
607     -0.875         0.875
95      -0.800         0.900
1154    -0.800         0.900
```

```
1022    -0.800         0.800
1163    -0.800         0.900
489     -0.800         0.900
```

In [26]:

```python
# Some examples of the most retweeted content.
print(df_tweets_keurig.sort_values(by='retweets', ascending=False).head(20
```

```
                                                text  retweets  favorite
s  \
1410  🇺🇸 I've used a Keurig for years and recently h...        61        57
0
419   My boss passed away unexpectedly. It’s been ro...        17        16
8
1374  Today I will be holding tutorials on how to sc...        17        37
4
219   Congrats to ~300 workers at Keurig Dr. Pepper ...        14         7
7
733   Keurig factory workers in Victorville vote to ...        14         2
3
666   Keurig Dr Pepper Inc. workers in Victorville h...        11         2
8
207   Shelby McEwen qualified 8th for Sunday's men's...        10         5
1
280   Keurig factory workers in Victorville vote to ...         6         1
6
916   came home from vacation to find all of our pet...         6         2
4
1179  Don’t you hate it when you forget to put a cof...         6         3
2
1592  I just submitted a Warranty claim for my broke...         5
3
844   Meet our newest and smartest brewer yet: the K...         3         1
1
1339  Keurig Mat. Coffee Pot Mat, Monogrammed, Dish ...         3
2
43    From cannabis Keurig pods to sodas, these are ...         3
7
566   Is Keurig Dr Pepper Stock Undervalued Even Aft...         3
0
186   Ummmm… no one learned from Keurig? https://t.c...         3         1
9
1524  I just submitted a Warranty claim for my broke...         3
6
914   Might I interest you gentleman in Keurig? Fast...         3         1
3
798   Can the DoNotPay claim my warranty for broken ...         3
5
803   After 550 days, the Keurig is rusty but (I bel...         3         6
8

                 user                                          text_clean  \
1410     Trumpeteer14  used keurig years recently realized much money...
419     Froschauer_AF  boss passed away unexpectedly rough dont want ...
1374    KelliGOPHATER  today holding tutorials screw cap back toothpa...
219   CalTeamstersPAC  congrats workers keurig pepper victorville vot...
733      VVDailyPress  keurig factory workers victorville vote unioni...
666      mattalley413  keurig dr pepper workers victorville voted uni...
207    JakeThompsonOE  shelby mcewen qualified 8th sunday men high ju...
280      steelworkers  keurig factory workers victorville vote unioni...
```

```
916      PeteGGriffin  came home vacation find pets wearing black tur...
1179        HarborMom        dont hate forget put coffee cup keurig uggg
1592      mypenplease  submitted warranty claim broken keurig k90 eli...
844            Keurig  meet newest smartest brewer yet plus smart bre...
1339  HandbagsByDyana  keurig mat coffee pot mat monogrammed dish dry...
43         Herbworthy        cannabis keurig pods sodas best weed drinks
566            fuzweb    keurig dr pepper stock undervalued even 80 jump
186            wbm312                           ummmm one learned keurig
1524      mypenplease  submitted warranty claim broken keurig k90 eli...
914         GingaHerb  might interest gentleman keurig fast tasty cof...
798       mypenplease  donotpay claim warranty broken keurig k90 elit...
803   LombardiHimself  550 days keurig rusty believe still works seas...

      polarity  subjectivity
1410  0.050000      0.129167
419   0.350000      0.575000
1374  0.000000      0.000000
219   0.400000      0.450000
733   0.000000      0.000000
666  -0.069444      0.583333
207   0.080000      0.770000
280   0.000000      0.000000
916  -0.205556      0.411111
1179 -0.800000      0.900000
1592  0.016667      0.616667
844   0.431412      0.711851
1339  0.000000      0.000000
43    1.000000      0.300000
566   0.000000      0.000000
186   0.000000      0.000000
1524  0.016667      0.616667
914   0.329167      0.779167
798  -0.400000      0.400000
803   0.700000      0.600000
```

In [27]:

```
# Top-20 users producing the most retweeted content.
print(df_tweets_keurig.sort_values(by='retweets', ascending=False).head(20
```

```
1410       Trumpeteer14
419       Froschauer_AF
1374      KelliGOPHATER
219     CalTeamstersPAC
733        VVDailyPress
666        mattalley413
207      JakeThompsonOE
280        steelworkers
916        PeteGGriffin
1179          HarborMom
1592        mypenplease
844              Keurig
1339    HandbagsByDyana
43           Herbworthy
566              fuzweb
186              wbm312
1524        mypenplease
914           GingaHerb
798         mypenplease
803     LombardiHimself
Name: user, dtype: object
```

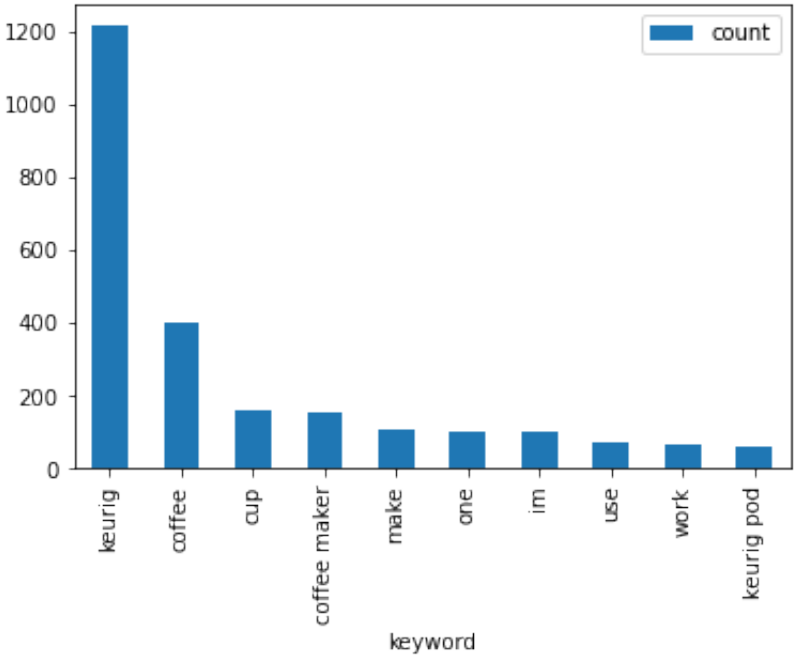
In [28]:

```
def brand_all_text(b):
    return ' '.join(df_tweets_keurig[df_tweets_keurig['text_clean'].str.co

# Most common content keywords for each brand with word cloud visualizatio
b = 'keurig'
wc = WordCloud(width=1200, height=800, max_font_size=110, collocations=Fal
plt.axis("off")
plt.imshow(wc, interpolation="bilinear")
plt.show()

kwords_keurig = WordCloud().process_text(brand_all_text(b))
df_kwords_keurig = pd.DataFrame(list(kwords_keurig.items()), columns=['key
print(df_kwords_keurig.sort_values(by = "count", ascending=False).head(10)
print(df_kwords_keurig.sort_values(by = "count", ascending=False).head(10)
```

```
               count
keyword
keurig          1215
coffee           400
cup              160
coffee maker     154
make             107
one              104
im               102
use               75
work              65
keurig pod        63
AxesSubplot(0.125,0.125;0.775x0.755)
```



In [33]:
```
tweets_keurig[1304]['full_text']
```

Out[33]: 'Buying this Keurig was the best thing ever, Ive had 3 cappuccinos already 😂😂'

In [34]:
```
tweets_keurig[740]['full_text']
```

Out[34]:
```
'yo yo the keurig lady on the phone today, i hope ur having a wonderful day
and ily thank u sm for helping me T-T'
```