In [1]:

```
import glob
import pandas as pd
import os
import matplotlib.pyplot as plt
import seaborn as sns
from os import path
import collections
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import seaborn as sns
from scipy.stats import norm
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import StandardScaler
from sklearn import metrics
from sklearn.metrics import accuracy_score
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
import string
import re
import nltk
from nltk import pos_tag
from nltk.corpus import stopwords
from nltk.tokenize import WhitespaceTokenizer
from nltk.stem import WordNetLemmatizer
from nltk.stem import PorterStemmer
import pandas as pd
from os import path
from PIL import Image
from wordcloud import WordCloud, STOPWORDS, ImageColorGenerator
```

In [2]:

```
cd excel_files/
```

```
[Errno 2] No such file or directory: 'excel_files/'
/Users/vishal/Desktop/DSPM Final Project
```

In [3]:

```
path = os.getcwd()
path
```

Out[3]:

```
'/Users/vishal/Desktop/DSPM Final Project'
```

In [4]:

```
all_data = pd.DataFrame()

for f in glob.glob(path+"/*.xlsx"):
    df = pd.read_excel(f)
    all_data = all_data.append(df,ignore_index=True)
```

How do people feel about price, quality, and value

In [ ]:

```

```

In [327]:

```
df_iphone_X_price = df_iphone_X[(df_iphone_X['sentence'].str.contains('price'))
| (df_iphone_X['sentence'].str.contains("cost"))]

print("Average polarity for Iphone X based on price")
print(df_iphone_X_price['polarity'].mean())

df_iphone_X_quality = df_iphone_X[(df_iphone_X['sentence'].str.contains('quality
')) | (df_iphone_X['sentence'].str.contains("quality"))]

print("Average polarity for Iphone X based on quality")
print(df_iphone_X_quality['polarity'].mean())

df_iphone_X_value = df_iphone_X[(df_iphone_X['sentence'].str.contains('value'))]

print("Average polarity for Iphone X based on value")
print(df_iphone_X_value['polarity'].mean())

iphonex = np.array([df_iphone_X_value['polarity'].mean() , df_iphone_X_price['po
larity'].mean() ,df_iphone_X_quality['polarity'].mean()  ])
```

```
Average polarity for Iphone X based on price
0.11693266592863485
Average polarity for Iphone X based on quality
0.13888553597028652
Average polarity for Iphone X based on value
0.08064625850340137
```

In [296]:

```
df_iphone_X_quality = df_iphone_X[(df_iphone_X['sentence'].str.contains('quality
')) | (df_iphone_X['sentence'].str.contains("quality"))]

print("Average polarity for Iphone X based on quality")
df_iphone_X_quality['polarity'].mean()
```

```
Average polarity for Iphone X based on quality
```

Out[296]:

```
0.13888553597028652
```

In [297]:

```
df_iphone_X_value = df_iphone_X[(df_iphone_X['sentence'].str.contains('value'))]

print("Average polarity for Iphone X based on value")
df_iphone_X_value['polarity'].mean()
```

```
Average polarity for Iphone X based on value
```

Out[297]:

```
0.08064625850340137
```

In [ ]:

```

```

In [271]:

```
df = df_iphone_X_price
att = []
for values in df['attr']:
    for pair in values :
        if(pair[1] == 'NN' or pair[1] == 'NNS' or pair[1] == 'NNP' or pair[1]=='
NNPS'):
            att.append(pair[0])
        if(pair[1] == 'JJ' or pair[1] == 'JJS' or pair[1] == 'JJR'):
            att.append(pair[0])
print("Most common attributes")
Counter(att).most_common(25)
```

Most common attributes

Out[271]:

```
[('gold', 92),
 ('silver', 89),
 ('grey', 87),
 ('new', 71),
 ('price', 63),
 ('ready', 53),
 ('iphone', 47),
 ('’', 44),
 ('•', 40),
 ('i', 39),
 ('jualanindo', 34),
 ('screen', 31),
 ('september', 30),
 ('cost', 28),
 ('phone', 25),
 ('iphones', 25),
 ('update', 23),
 ('iphone8plus', 23),
 ('open', 23),
 ('di', 22),
 ('pricelist', 22),
 ('po', 21),
 ('november', 21),
 ('akan', 21),
 ('sekitar', 21)]
```

In [ ]:

```
```

In [277]:

```
df_galaxy_price = df_galaxy[(df_galaxy['sentence'].str.contains('price')) | (df_
galaxy['sentence'].str.contains("cost"))]

df_galaxy_price['polarity'].mean()
```

Out[277]:

```
0.16597004584035197
```

In [292]:

```
df = df_galaxy_price
att = []
for values in df['attr']:
    for pair in values :
        if(pair[1] == 'NN' or pair[1] == 'NNS' or pair[1] == 'NNP' or pair[1]=='
NNPS'):
            att.append(pair[0])
        if(pair[1] == 'JJ' or pair[1] == 'JJS' or pair[1] == 'JJR'):
            att.append(pair[0])
print("Most common attributes")
Counter(att).most_common(25)
```

```
Most common attributes
```

Out[292]:

```
[('galaxy', 70027),
 ('samsung', 48494),
 ('’', 22485),
 ('phone', 15582),
 ('s8+', 15426),
 ('new', 13449),
 ('price', 12164),
 ('screen', 9371),
 ('camera', 7509),
 ('display', 7218),
 ('phones', 6574),
 ('i', 6172),
 ('smartphone', 6149),
 ('available', 5597),
 ('battery', 5320),
 ('device', 5286),
 ('case', 5247),
 ('s7', 4968),
 ('android', 4937),
 ('note', 4777),
 ('flagship', 4720),
 ('best', 4487),
 ('features', 4138),
 ('devices', 4117),
 ('smartphones', 4112)]
```

In [293]:

```
print("Average polarity for Galaxy based on price")
df['polarity'].mean()
```

```
Average polarity for Galaxy based on price
```

Out[293]:

```
0.16597004584035197
```

In [284]:

```
df_galaxy_quality = df_galaxy[(df_galaxy['sentence'].str.contains("quality"))]
```

In [285]:

```
df = df_galaxy_quality
att = []
for values in df['attr']:
    for pair in values :
        if(pair[1] == 'NN' or pair[1] == 'NNS' or pair[1] == 'NNP' or pair[1]=='
NNPS'):
            att.append(pair[0])
        if(pair[1] == 'JJ' or pair[1] == 'JJS' or pair[1] == 'JJR'):
            att.append(pair[0])
print("Most common attributes")
Counter(att).most_common(25)
```

Most common attributes

Out[285]:

```
[('galaxy', 27976),
 ('samsung', 18151),
 ('’', 9449),
 ('phone', 7721),
 ('s8+', 6837),
 ('screen', 6355),
 ('new', 5279),
 ('quality', 5113),
 ('case', 4569),
 ('camera', 4499),
 ('i', 4381),
 ('display', 3903),
 ('design', 2917),
 ('battery', 2813),
 ('phones', 2693),
 ('best', 2462),
 ('smartphone', 2305),
 ('device', 2222),
 ('devices', 2184),
 ('android', 2021),
 ('features', 1929),
 ('glass', 1929),
 ('protector', 1927),
 ('note', 1910),
 ('s7', 1904)]
```

In [287]:

```
print("Average polarity for Galaxy based on quality")
df['polarity'].mean()
```

Average polarity for Galaxy based on quality

Out[287]:

0.20861209267627492

In [308]:

```
galaxy = np.array([df_galaxy_value['polarity'].mean(),df_galaxy_cost['polarity']
.mean() , df_galaxy_quality['polarity'].mean()])
```

In [326]:

```
#galaxy =  np.array(list)

df_galaxy_value = df_galaxy[(df_galaxy['sentence'].str.contains("value"))]
print("Average polarity for Galaxy based on value")
#galaxy.add(df_galaxy_value['polarity'].mean())
print(df_galaxy_value['polarity'].mean())

df_galaxy_cost = df_galaxy[(df_galaxy['sentence'].str.contains("cost")) |(df_gal
axy['sentence'].str.contains("price")) ]
print("Average polarity for Galaxy based on cost")
#galaxy.add(df_galaxy_cost['polarity'].mean())
print(df_galaxy_cost['polarity'].mean())

df_galaxy_quality = df_galaxy[(df_galaxy['sentence'].str.contains("quality"))]
print("Average polarity for Galaxy based on quality")
#galaxy.add(df_galaxy_quality['polarity'].mean())
print(df_galaxy_quality['polarity'].mean())
```

```
Average polarity for Galaxy based on value
0.17678088108756287
Average polarity for Galaxy based on cost
0.16597004584035197
Average polarity for Galaxy based on quality
0.20861209267627492
```

In [289]:

```
df = df_galaxy_value
att = []
for values in df['attr']:
    for pair in values :
        if(pair[1] == 'NN' or pair[1] == 'NNS' or pair[1] == 'NNP' or pair[1]=='
NNPS'):
            att.append(pair[0])
        if(pair[1] == 'JJ' or pair[1] == 'JJS' or pair[1] == 'JJR'):
            att.append(pair[0])
print("Most common attributes")
Counter(att).most_common(25)
```

Most common attributes

Out[289]:

```
[('galaxy', 12679),
 ('samsung', 7959),
 ('’', 3857),
 ('phone', 3387),
 ('screen', 3298),
 ('s8+', 2831),
 ('new', 2021),
 ('i', 1873),
 ('case', 1788),
 ('protector', 1496),
 ('value', 1491),
 ('best', 1362),
 ('camera', 1251),
 ('phones', 1234),
 (']', 1164),
 ('product', 1156),
 ('android', 1125),
 ('s7', 1083),
 ('[', 1008),
 ('display', 996),
 ('glass', 972),
 ('device', 920),
 ('design', 909),
 ('”', 867),
 ('full', 853)]
```

In [290]:

```
print("Average polarity for Galaxy based on value")
df['polarity'].mean()
```

Average polarity for Galaxy based on value

Out[290]:

0.17678088108756287

In [ ]:

In [319]:

```
df = df_iphone_8
df['Sound Bite Text'].apply(lambda x: x.lower())
df['tokenized_text'] = df['Sound Bite Text'].apply(word_tokenize)
df['tokenized_text'] = df['tokenized_text'].apply(lambda x: [item for item in x
if item not in stop_words])
#df['tokenized_text'] = df['tokenized_text'].apply(lambda x: [lemmatizer.lemmati
ze(y) for y in x])
#stemmer = PorterStemmer()
#df['tokenized_text'] = df['tokenized_text'].apply(lambda x: [stemmer.stem(y) fo
r y in x])
df['tokenized_text'] = df['tokenized_text'].apply(lambda x: [item.lower() for it
em in x])
df['sentence'] = df['tokenized_text'].apply(' '.join)
df['attr'] = df['tokenized_text'].apply(lambda x: nltk.pos_tag(x))
att = []
for values in df['attr']:
    for pair in values :
        if(pair[1] == 'NN' or pair[1] == 'NNS' or pair[1] == 'NNP' or pair[1]=='
NNPS'):
            att.append(pair[0])
        if(pair[1] == 'JJ' or pair[1] == 'JJS' or pair[1] == 'JJR'):
            att.append(pair[0])
print("Most common attributes")
Counter(att).most_common(25)
```

Most common attributes

Out[319]:

```
[('new', 61085),
 ('’', 60916),
 ('i', 50642),
 ('phone', 27763),
 ('camera', 21705),
 ('tags', 15511),
 ('ios', 15222),
 ('https', 15174),
 ('design', 14278),
 ('case', 14155),
 ('screen', 13679),
 ('year', 13563),
 ('display', 13135),
 ('news', 12961),
 ('device', 12194),
 ('wireless', 11740),
 ('”', 11701),
 ('—', 11168),
 ('smartphone', 10756),
 ('plus', 10754),
 ('time', 10682),
 ('“', 10266),
 ('video', 9876),
 ('first', 9836),
 ('technology', 9767)]
```

In [321]:

```
df_iphone_8.head()
```

Out[321]:

	Post ID	Sound Bite Text	Ratings and Scores	Title	Source Type	Post Type	Media Type	
30033	8.36905e+17	Another small teaser (A) iPhone 8 #iPhone8 #iP...	NaN	NaN	Twitter	Original	Image	http://twitter.com/C
62726	8.3694e+17	Instagram Media from: the.luxurygram, New iPho...	NaN	NaN	Twitter	Original	No Media	http://twitter.com/ibr
62752	8.36941e+17	iPhone 8 To Ditch Lightning Port In Favor Of U...	NaN	NaN	Twitter	Original	Link	http://twitter.com/It
35346	8.36929e+17	iPhone 8 to Sport Fingerprint Scanner Undernea...	NaN	NaN	Twitter	Original	Image; Link	http://twitter.com/ga
63207	8.36927e+17	iPhone 8 to use USB-C? Xbox to subscription ga...	NaN	NaN	Twitter	Original	No Media	http://twitter.com/h

In [ ]:

In [325]:

```
df_iphone_8_price = df_iphone_8[(df_iphone_8['sentence'].str.contains('price'))
| (df_iphone_8['sentence'].str.contains("cost"))]

print("Average polarity for Iphone X based on price")
print(df_iphone_8_price['polarity'].mean())

df_iphone_8_quality = df_iphone_8[(df_iphone_8['sentence'].str.contains('quality
')) | (df_iphone_8['sentence'].str.contains("quality"))]

print("Average polarity for Iphone X based on quality")
print(df_iphone_8_quality['polarity'].mean())

df_iphone_8_value = df_iphone_8[(df_iphone_8['sentence'].str.contains('value'))]

print("Average polarity for Iphone X based on value")
print(df_iphone_8_value['polarity'].mean())

iphone8 = np.array([df_iphone_8_value['polarity'].mean() , df_iphone_8_price['po
larity'].mean() ,df_iphone_8_quality['polarity'].mean()  ])
```

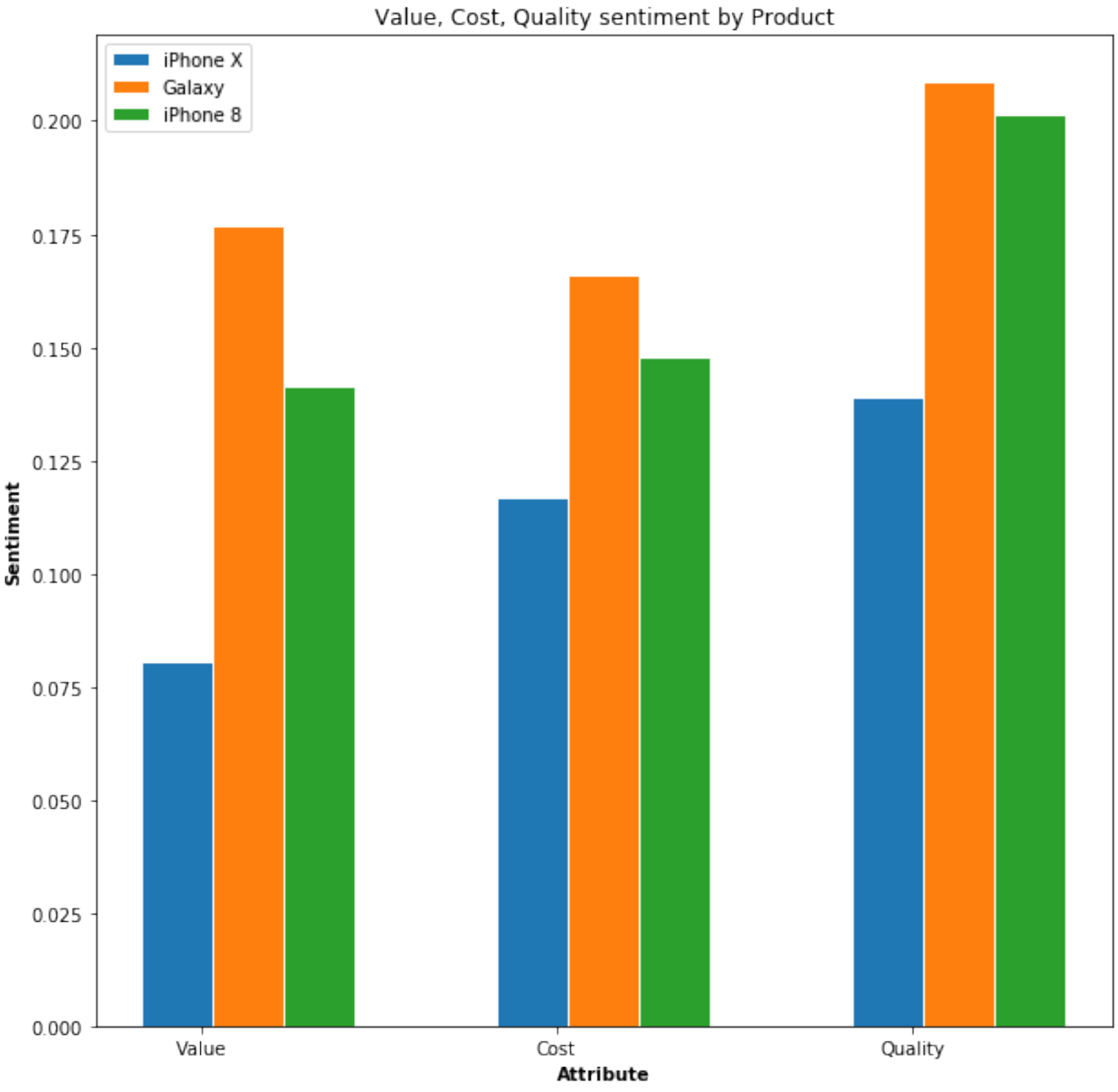
```
Average polarity for Iphone X based on price
0.147805610001331
Average polarity for Iphone X based on quality
0.20154193868620604
Average polarity for Iphone X based on value
0.14154775545655 32
```

In [363]:

```
barWidth = 0.2
x1 = np.arange(len(iphonex))
x2 = [x + barWidth for x in x1]
x3 =[x + barWidth +barWidth for x in x1]
plt.figure(figsize=(10,10))
plt.bar(x1, iphonex, color='C0', width=barWidth, edgecolor='white', label='iPhon
e X')
plt.bar(x2, galaxy, color='C1', width=barWidth, edgecolor='white', label='Galaxy
')
plt.bar(x3, iphone8, color='C2', width=barWidth, edgecolor='white', label='iPhon
e 8')
plt.xlabel('Attribute', fontweight='bold')
plt.ylabel('Sentiment', fontweight='bold')
plt.xticks([r + barWidth/3 for r in range(len(galaxy))], ['Value', 'Cost','Quali
ty'])
plt.title('Value, Cost, Quality sentiment by Product')
plt.legend()
```

Out[363]:

<matplotlib.legend.Legend at 0x1c37800410>



Above graph shows average polarity towards Price, Quality, and Value