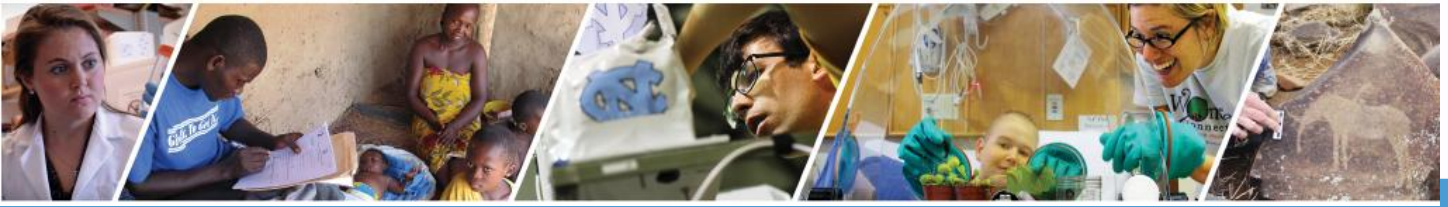
Project Report

Predicting if the Customer's Review is Negative or Positive using NLP



Presenter: Yiran Li

UNC RESEARCH

1. Start with a Question

2. Data Gathering + Data Cleaning

3. Model Selection and Further Analysis

4. Future Directions

Can we tell how well Amazon is doing with their customers?

We want to know if the customer's review is positive or negative.

1. Start with a Question

2. Data Gathering + Data Cleaning

3. Model Selection and Further Analysis

4. Future Directions

Dataset we are using:

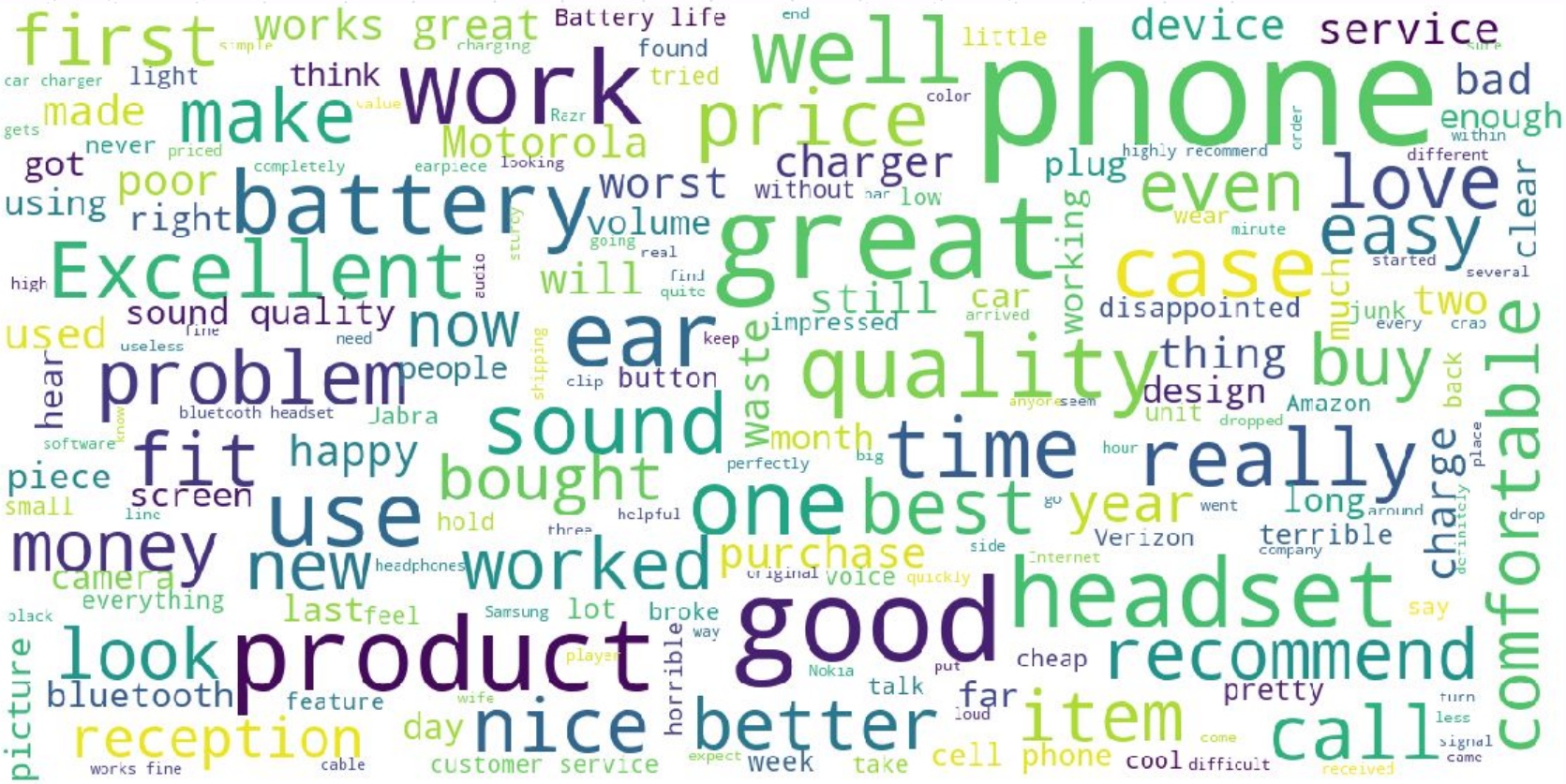
Labeled customers' reviews of Amazon

1 = positive review

0 = negative review

So there is no way for me to plug it in here in the US unless I go by a converter.	0
Good case, Excellent value.	1
Great for the jawbone.	1
Tied to charger for conversations lasting more than 45 minutes.MAJOR PROBLEMS!!	0
The mic is great.	1
I have to jiggle the plug to get it to line up right to get decent volume.	0
If you have several dozen or several hundred contacts, then imagine the fun of sending each of them one by one.	0
If you are Razr owner...you must have this!	1
Needless to say, I wasted my money.	0
What a waste of money and time!.	0

Word Cloud


first
works
great
Battery life
end
well
little
device
service
phone
work
found
tried
think
light
car charger
bad
made
make
price
Motorola
enough
never
priced
completely
earpiece
looking
charger
plug
highly recommend
different
got
poor
worst
without
low
even
love
using
right
battery
volume
easy
clear
Excellent
great
case
will
still
car
arrived
disappointed
junk
two
sound quality
now
ear
impressed
quality
thing
buy
used
useless
need
keep
waste
working
much
every
crap
hear
problem
people
button
clip
shipping
design
Amazon
unit
dropped
back
bluetooth headset
Jabra
sound
month
time
really
charge
software
fit
happy
bought
perfectly
big
hour
piece
screen
small
one
best
year
went
long
around
drop
use
hold
three
helpful
Verizon
terrible
company
money
new
worked
purchase
side
Internet
camera
headphones
original
voice
quickly
everything
last
feel
Samsung
lot
broke
way
say
headset
good
comfortable
look
product
horrible
cheap
recommend
bluetooth
feature
wife
Nokia
talk
far
pretty
picture
reception
day
nice
better
item
call
cell
phone
cool
difficult
works fine
cable
customer service
expect
week
take
received
signal
come

Import the data

```
import pandas as pd

filepath_dict={'amazon': '/Users/amy/Downloads/amazon_cells_labelled.txt'
               }
df_list=[]

for source,filepath in filepath_dict.items():
    df=pd.read_csv(filepath, names=['sentence','label'], sep='\t')
    df['source']=source
    df_list.append(df)

df=pd.concat(df_list)
print(df.iloc[0])
```

```
sentence    So there is no way for me to plug it in here i...
label                                                       0
source                                                 amazon
Name: 0, dtype: object
```

```
df_amazon=df[df['source']=='amazon']
```

```
df_amazon.head(10)
```

	sentence	label	source
0	So there is no way for me to plug it in here i...	0	amazon
1	Good case, Excellent value.	1	amazon
2	Great for the jawbone.	1	amazon
3	Tied to charger for conversations lasting more...	0	amazon
4	The mic is great.	1	amazon
5	I have to jiggle the plug to get it to line up...	0	amazon
6	If you have several dozen or several hundred c...	0	amazon
7	If you are Razr owner...you must have this!	1	amazon
8	Needless to say, I wasted my money.	0	amazon
9	What a waste of money and time!.	0	amazon

Lowercase the texts

```
#lower case
df_amazon['sentence']=df_amazon['sentence'].str.lower()
```

```
df_amazon.head(10)
```

	sentence	label	source
0	so there is no way for me to plug it in here i...	0	amazon
1	good case, excellent value.	1	amazon
2	great for the jawbone.	1	amazon
3	tied to charger for conversations lasting more...	0	amazon
4	the mic is great.	1	amazon
5	i have to jiggle the plug to get it to line up...	0	amazon
6	if you have several dozen or several hundred c...	0	amazon
7	if you are razr owner...you must have this!	1	amazon
8	needless to say, i wasted my money.	0	amazon
9	what a waste of money and time!.	0	amazon

Remove punctuations and special characters

```
df_amazon['sentence']=df_amazon['sentence'].str.replace("[^\w\s]",'')
df_amazon.head(10)
```

	sentence	label	source
0	so there is no way for me to plug it in here i...	0	amazon
1	good case excellent value	1	amazon
2	great for the jawbone	1	amazon
3	tied to charger for conversations lasting more...	0	amazon
4	the mic is great	1	amazon
5	i have to jiggle the plug to get it to line up...	0	amazon
6	if you have several dozen or several hundred c...	0	amazon
7	if you are razr owneryou must have this	1	amazon
8	needless to say i wasted my money	0	amazon
9	what a waste of money and time	0	amazon

Removing Stop Words; Stemming and Lemmatization

```
import nltk
nltk.download('stopwords')
from nltk.stem import WordNetLemmatizer
from nltk.stem.porter import *
from nltk.corpus import stopwords
print(stopwords.words('english'))

lemmatizer=WordNetLemmatizer()
stemmer=PorterStemmer()
stop_words=stopwords.words('english')

def custom_tokenize(text,stopwords):
    text=str(text).split()
    #text =[t for t in text if t not in stopwords]
    text =[stemmer.stem(lemmatizer.lemmatize(t)) for t in text if t not in stopwords and len(t)>2]
    text=" ".join(text[0:])
    return(text)
```

Removing Stop Words; Stemming and Lemmatization

```
df_amazon['sentence_clean']=df_amazon.sentence.apply(lambda x: custom_tokenize(x,stop_words))
```

```
df_amazon.head(10)
```

	sentence	label	source	sentence_clean
0	so there is no way for me to plug it in here i...	0	amazon	way plug unless converter
1	good case excellent value	1	amazon	good case excellent value
2	great for the jawbone	1	amazon	great jawbone
3	tied to charger for conversations lasting more...	0	amazon	tied charger conversations lasting minutesmajo...
4	the mic is great	1	amazon	mic great
5	i have to jiggle the plug to get it to line up...	0	amazon	jiggle plug get line right get decent volume
6	if you have several dozen or several hundred c...	0	amazon	several dozen several hundred contacts imagine...
7	if you are razr owneryou must have this	1	amazon	razr owneryou must
8	needless to say i wasted my money	0	amazon	needless say wasted money
9	what a waste of money and time	0	amazon	waste money time

Training & Testing set

```
from sklearn.model_selection import train_test_split
sentences=df_amazon['sentence'].values
y=df_amazon['label'].values
sentences_train, sentences_test, y_train, y_test=train_test_split(sentences,y,test_size=0.25,random_state=323)

from sklearn.feature_extraction.text import TfidfVectorizer
vec=TfidfVectorizer(#stop_words='english',
                    token_pattern=r'\w{2,}', # only keeps the words with 2 letters or more
                    max_features=2000,
                    ngram_range=(1,3), # unigram, bigrams and trigrams
                    norm='l2') # not important
vec.fit(sentences_train)
tfidf_features=vec.get_feature_names()
#tfidf_features

x_train=vec.transform(sentences_train)
x_test=vec.transform(sentences_test)
```

1. Start with a Question

2. Data Gathering + Data Cleaning

3. Model Selection and Further Analysis

4. Future Directions

Apply Logistic Regression

```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression(solver='lbfgs')
lr.fit(x_train,y_train)
predict_lr = lr.predict(x_test)
score = lr.score(x_test,y_test)
result_lr = pd.DataFrame({'Predict':predict_lr,'actual':y_test})
print(f'Accuracy: {score}')
```

```
Accuracy: 0.812
```

```
from sklearn import metrics
from sklearn.metrics import classification_report #commonly used
from sklearn.metrics import confusion_matrix #also very popular
clr = classification_report(y_test,predict_lr)
com = confusion_matrix(y_test,predict_lr)
print(clr)
print("Confusion_matrix")
print(com)
```

```
              precision    recall  f1-score   support

           0       0.80      0.84      0.82       129
           1       0.82      0.78      0.80       121

    accuracy                           0.81       250
   macro avg       0.81      0.81      0.81       250
weighted avg       0.81      0.81      0.81       250

Confusion_matrix
[[109  20]
 [ 27  94]]
```

Apply Random Forest

```
import sklearn
import sklearn.ensemble
#Acuuracy
from sklearn.metrics import classification_report,confusion_matrix
rf=sklearn.ensemble.RandomForestClassifier(n_estimators=500, random_state=323)
rf.fit(x_train, y_train)
pred=rf.predict(x_test)
pred_proba=rf.predict_proba(x_test)
confusionm = confusion_matrix(y_test,pred)
print(classification_report(y_test,pred))
print("Confusion_matrix")
print(confusionm)
```

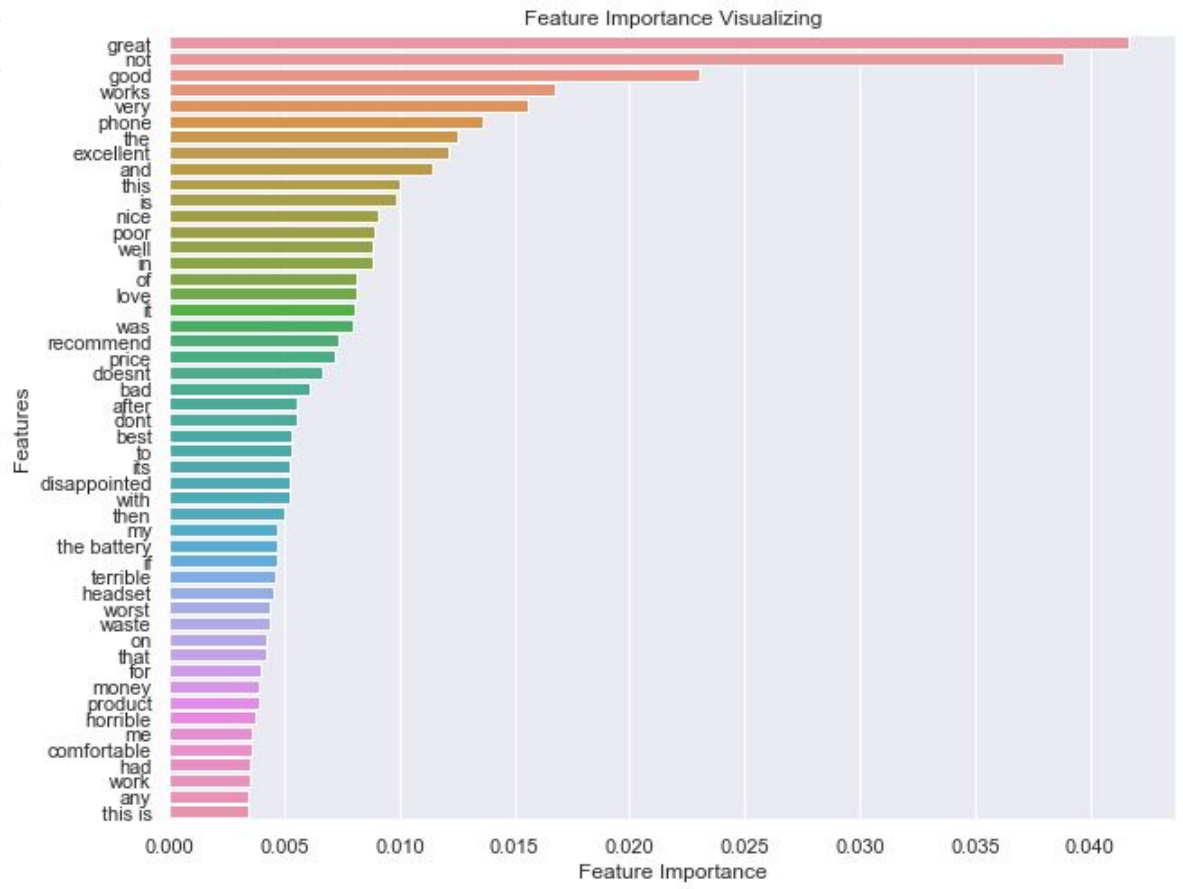
```
              precision    recall  f1-score   support

           0       0.79      0.81      0.80       129
           1       0.79      0.77      0.78       121

    accuracy                           0.79       250
   macro avg       0.79      0.79      0.79       250
weighted avg       0.79      0.79      0.79       250

Confusion_matrix
[[105  24]
 [ 28  93]]
```

Visualizing the Feature Importance



UNC RESEARCH

Apply XGBoost

```
import xgboost as xgb
mdl=xgb.XGBClassifier(learning_rate=0.05,
                      n_estimators=500,
                      max_depth=3,
                      subsample=0.8,
                      objective='binary:logistic',
                      monotone_constraints="1",
                      nthread=-1)
mdl.fit(x_train, y_train)

from sklearn.metrics import classification_report,confusion_matrix
pred=mdl.predict(x_test)
pred_proba=mdl.predict_proba(x_test)
confusionmat = confusion_matrix(y_test,pred)
print(classification_report(y_test,pred))
print("Confusion_matrix")
print(confusionmat)
```

```
              precision    recall  f1-score   support

           0       0.73      0.84      0.78       129
           1       0.80      0.67      0.73       121

    accuracy                           0.76       250
   macro avg       0.77      0.76      0.76       250
weighted avg       0.77      0.76      0.76       250

Confusion_matrix
[[109  20]
 [ 40  81]]
```

1. Start With a Question

2. Data Gathering + Data Cleaning

3. Model Selection and Further Analysis

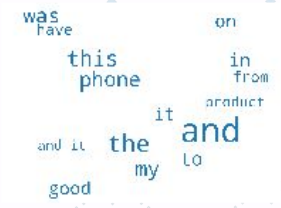
4. Further Directions

Further Directions

Apply Topic Modeling:

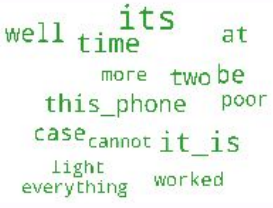
To see what topics has been said in each customer review, i.e. what's the bag of words in each topic.

Topic 0

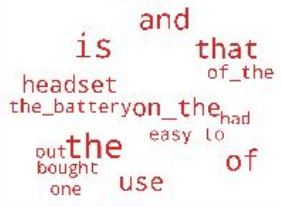


Topic 1

Topic 2

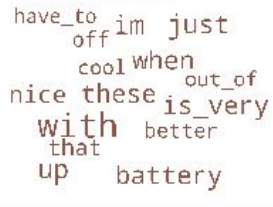


Topic 3

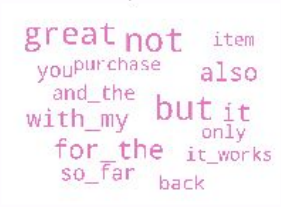


Topic 4

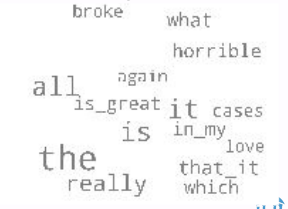
Topic 5



Topic 6



Topic 7



Topic 8

Questions?

Thank you!

Yiran Li
yiran1@live.unc.edu