



รายงานประจำวิชา

เรื่องเฉพาะทางวิทยาการคอมพิวเตอร์

(Select Topic in Computer Science)

รหัสวิชา 01418496 หมู่ 800

เรื่อง

Sentiment Analysis

จัดทำโดย

สมาชิกกลุ่ม

- | | | |
|------------------|--------------|----------------------|
| 1. นางสาวภาณุมาศ | ชาติโรสง | รหัสนิสิต 6530200347 |
| 2. นางสาวมนิษฐา | เบญจเจดสิริ | รหัสนิสิต 6530200380 |
| 3. นางสาวอักษรา | สุวัณณะวัยค์ | รหัสนิสิต 6530200886 |
| 4. นายอภูติน | สาละ | รหัสนิสิต 6530200894 |

เสนอ

อาจารย์อรรณ อิมสมบัติ

1. ที่มา และ วัตถุประสงค์

แบรนด์เครื่องสำอางมีความสำคัญในอุตสาหกรรมเครื่องสำอางและความงามมุ่งสู่การสรรหาผลิตภัณฑ์ที่ดีและใช้งานได้ง่าย เพื่อรองรับความต้องการของผู้บริโภคและความพึงพอใจในสินค้า ความคิดเห็นของลูกค้าเป็นสิ่งสำคัญที่ช่วยให้แบรนด์เครื่องสำอางเข้าใจความพึงพอใจของลูกค้าและมีโอกาสปรับปรุงผลิตภัณฑ์หรือบริการของพวกเขาให้ดียิ่งขึ้น

ในยุคที่สื่อสังคมและการแสดงความคิดเห็นในสถานการณ์ต่าง ๆ มีความสำคัญอย่างมาก การวิเคราะห์ความคิดเห็นเกี่ยวกับแบรนด์เครื่องสำอางในสื่อสังคมและแหล่งข้อมูลออนไลน์เป็นอุปกรณ์ที่มีประโยชน์อย่างมากในการหาความรู้สึกและความคิดเห็นของลูกค้าต่อผลิตภัณฑ์ของบริษัท การทำ Sentiment Analysis ในกรณีนี้มุ่งเน้นที่การวิเคราะห์ความคิดเห็นและความรู้สึกของผู้บริโภคเกี่ยวกับแบรนด์เครื่องสำอางเพื่อ

1. เข้าใจความพึงพอใจและความไม่พึงพอใจของลูกค้าต่อผลิตภัณฑ์ของแบรนด์
2. ตรวจสอบความนิยมของแบรนด์เครื่องสำอางในสื่อสังคมและแหล่งข้อมูลออนไลน์

การวิเคราะห์ความคิดเห็นเกี่ยวกับแบรนด์เครื่องสำอางจะช่วยให้แบรนด์เครื่องสำอางทราบถึงความพึงพอใจและความคิดเห็นของลูกค้าต่อผลิตภัณฑ์ของพวกเขาและสร้างโอกาสในการปรับปรุงผลิตภัณฑ์และบริการของพวกเขาอย่างมีประสิทธิภาพเพื่อตอบสนองความต้องการของลูกค้า

2. Dataset

- Link google sheet : https://docs.google.com/spreadsheets/d/1O-mJU-hzp7eln7hFbVZbnS3rBzX0EPgmBNSHH_acalg/edit?usp=sharing
- จำนวนข้อมูล
 - ความคิดเห็นเชิงบวก 50 ความคิดเห็น
 - ความคิดเห็นเชิงปกติ 50 ความคิดเห็น
 - ความคิดเห็นเชิงลบ 51 ความคิดเห็น

3. กระบวนการ pre-processing ต่างๆ ที่ทำ

- อ่านไฟล์

```
data = pd.read_csv("/content/drive/MyDrive/Colab Notebooks/image/sentiment_lagallce.csv")  
print(data.head(20))
```

- แปลงคอลัมน์ class ให้เป็น ตัวเลขที่กำหนด

```
131 sentiment_mapping = {'positive': 2, 'neutral': 1, 'negative': 0}  
data['class'] = data['class'].replace(sentiment_mapping)  
print(data.head(20))
```

	class	sentence
0	2	เบบี๋ลัษไช้ดีมาก ทาแล้วแก้มชมพูน่ารัก
1	2	เราชอบมาก น่ารักมาก ๆ เลยค่ะ หมดเมื่อไหร่กลับมา...
2	2	เลิศเกิน สีสวยมาก
3	2	เห็นแพคเกจในคลิปรีวว่ากร๊าดแล้ว เห็นของจริง...

- ตัดคำ + train test ข้อมูล

แบ่งชุดข้อมูล X กับ y เพื่อเทรน

```
[ ] sentence = data['sentence'].astype(str)  
y = data['class'].values
```

ตัดคำ และ train test

```
[ ] X_tokenized = sentence.apply(word_tokenize, keep_whitespace=False)  
sentence_train, sentence_test, y_train, y_test = train_test_split(X_tokenized, y, test_size=0.3, random_state=100)
```

- แปลงข้อมูล (word representation) โดยใช้ countVectorizer()

```
[ ] vectorizer = CountVectorizer(analyzer=lambda x: x) # Use the list of tokens as the analyzer  
vectorizer.fit(sentence_train)  
  
#print(X_bow.shape) # (documents, vocab)
```

CountVectorizer
CountVectorizer(analyzer=<function <lambda> at 0x7def72a8a290>)

```
X_train= vectorizer.fit_transform(sentence_train)  
X_test = vectorizer.transform(sentence_test)
```

4. ผลการทดลองด้วย อัลกอริทึมต่างๆ, % การ train-test และ link ไฟล์ google colab

- ทดลองใช้ classification --> LogisticRegression()

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score

# Step 1: Train the binomial logistic regression
logreg_classifier = LogisticRegression()
logreg_classifier.fit(X_train, y_train)

# Step 2: Evaluate the model
y_pred = logreg_classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

Accuracy: 0.9130434782608695

- ทดลองใช้ classification --> KNN

```
[141] from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors=5, metric='minkowski', p=2)
classifier.fit(X_train, y_train)
```

▼ KNeighborsClassifier
KNeighborsClassifier()

```
y_pred = classifier.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

Accuracy: 0.717391304347826

- ทดลองใช้ classification --> MultinomialNB()

```
[143] from sklearn.naive_bayes import MultinomialNB

classifier_NB = MultinomialNB()
classifier_NB.fit(X_train, y_train)
```

▼ MultinomialNB
MultinomialNB()

```
[144] y_predict = classifier_NB.predict(X_test)
score = classifier_NB.score(X_test, y_test)
print("Accuracy:", score)
```

Accuracy: 0.6739130434782609

(https://colab.research.google.com/drive/1l1F2bvePzpHqU6k8zqMiYFc-KWE_DmCd?usp=sharing)

5. confusion matrix การจำแนก class ต่างๆ ของอัลกอริทึมที่มีผลสูงสุด

- confusion matrix เลือกใช้ classification --> LogisticRegression()

```
from sklearn.metrics import classification_report, confusion_matrix
# Print classification report (precision, recall, F1-score, support)
print("Classification Report:")
print(classification_report(y_test, y_pred))

# Print confusion matrix
print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
```

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.83	0.91	18
1	0.83	0.91	0.87	11
2	0.89	1.00	0.94	17
accuracy			0.91	46
macro avg	0.91	0.91	0.91	46
weighted avg	0.92	0.91	0.91	46

Confusion Matrix:

```
[[15  2  1]
 [ 0 10  1]
 [ 0  0 17]]
```

6. QR code Chatbot :

