## **Big Data in Media Technology**

**Lab 1: Sentiment Classification**

**with Naïve Bayesian Classifier (NBC) and SVM**



**Group 3**

**Name：Yangle Ma**

**Systematic Diagram**

Prepared Data（train、test、evaluation）

Imort liabries and data

Fit model in the Evaluation set

**Model**

**Evaluation**

**Data**

**Concat**

**Model**

**Building**

SVM

Naïve Bayesian Classifier

Remove repeated characters

Remove Emojis

Text lower

Mapping 0:’Negative’ 1: ‘Positive’

Remove HTML tags

Remove Hashtags、Mentions and URL

**Data**

**Cleaning**

**Feature Extraction**

1. Remove HTML tags

# https://beautiful-soup-4.readthedocs.io/en/latest/index.html?highlight=get\_text#get-text

    soup = BeautifulSoup(text, 'html.parser')

    text = soup.get\_text(strip=True)

1. Remove Hashtags、Mentions and URL

# https://docs.python.org/3/library/re.html#functions

    text = re.sub(r'#\S+', '', text)

    text = re.sub(r'@\S+', '', text)

    text = re.sub(r'http\S+', '', text)

1. Remove repeated characters

    text = re.sub(r'(.)\1+', r'\1', text)

1. Remove Emojis

#https://carpedm20.github.io/emoji/docs/#replacing-and-removing-emoji

    text = emoji.replace\_emoji(text, replace='')

1. Text lower and Remove brackets

    text = text.lower()

    return text.strip()

1. Apply the function

all\_data['text']=all\_data['text'].apply(preprocessing)

1. Mapping 0 : ‘Negative’ , 1 : ‘Positive’

#Transform 0 to negative，1 to positive

mapping = {

    0: 'negative',

    1: 'positive',

}

all\_data['score'] = all\_data['score'].map(mapping)

**Test**

1. **Naïve Bayesian Classifier**

# vectorizer = CountVectorizer()

vectorizer = TfidfVectorizer()

X = vectorizer.fit\_transform(train['text'])

X1 = vectorizer.transform(test['text'])

y = train['score']

y1 = test['score']

# Train the model

nb\_model = MultinomialNB(alpha=0.3)

nb\_model.fit(X, y)

# Predict and evaluate

y\_pred = nb\_model.predict(X1)

accuracy = accuracy\_score(y1, y\_pred)

print(f"Model Accuracy: {accuracy \* 100:.2f}%")

**Model Accuracy: 83.00%**

1. **SVM**

from sklearn.svm import SVC

svm\_model = SVC(kernel='linear')

svm\_model.fit(X, y)

y\_pred = svm\_model.predict(X1)

accuracy = accuracy\_score(y1, y\_pred)

print(f"Model Accuracy: {accuracy \* 100:.2f}%")

**Model Accuracy: 84.72%**

**Evaluation Results**

**1.Naïve Bayesian Classifier**

# Naïve Bayesian Evaluation

X2 = vectorizer.transform(evaluation['text'])

y2 = evaluation['score']

y\_pred1 = nb\_model.predict(X2)

accuracy1 = accuracy\_score(y2, y\_pred1)

print(f"Model Accuracy: {accuracy1 \* 100:.2f}%")

**Model Accuracy: 82.60%**

**2.SVM**

# SVM Evaluation

y\_pred1 = svm\_model.predict(X2)

accuracy1 = accuracy\_score(y2, y\_pred1)

print(f"Model Accuracy: {accuracy1 \* 100:.2f}%")

**Model Accuracy: 85.02%**

**References**

1. **<https://beautiful-soup-4.readthedocs.io/en/latest/index.html?highlight=get_text#get-text>**

**2. <https://docs.python.org/3/library/re.html#functions>**

1. **<https://carpedm20.github.io/emoji/docs/#replacing-and-removing-emoji>**
2. **<https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#sklearn.naive_bayes.MultinomialNB>**
3. **<https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC>**