



รายงานประจำวิชา

เรื่องเฉพาะทางวิทยาการคอมพิวเตอร์

(Selected Topic in Computer Science)

รหัสวิชา 01418496 หมู่เรียน 800

เรื่อง imdb _master

จัดทำโดย

รายชื่อสมาชิก

1. นางสาวปาณิสรา วิจารณ์ รหัสประจำตัวนิสิต 6530200274
2. นางสาวเพ็ญพิชชา ไพรวลย์ รหัสประจำตัวนิสิต 6530200312
3. นางสาวมนัสวี ปิยะโสภาสกุล รหัสประจำตัวนิสิต 6530200371
4. นางสาวบุญพิทักษ์ ผมเพชร รหัสประจำตัวนิสิต 6530200681
5. นางสาวเพชรรัตน์ ทองล้วน รหัสประจำตัวนิสิต 6530200746

เสนอ

อาจารย์ชโลธร ชูทอง

อาจารย์ประจำวิชา เรื่องเฉพาะทางวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์ ศรีราชา

มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขตศรีราชา

1. ลักษณะของข้อมูล

- Dataset: https://drive.google.com/file/d/1jOhr4yMjPPmkcAx8eAQpCytKnbtz8TH-/view?usp=drive_link
- เป็นชุดข้อมูลที่รวบรวมความคิดเห็นในการรีวิวหนังทั้งหมด 100,000 รายการ
- มี Label เป็นการแสดงความเห็นเชิงต่างๆ
 - Positive หรือ pos คือการแสดงความคิดเห็นเชิงบวก
 - Negative หรือ neg คือการแสดงความคิดเห็นเชิงลบ
 - Unsupervised หรือ unsup คือการแสดงความคิดเห็นกลางๆที่ไม่ไปเชิงบวกหรือเชิงลบมากเกินไป

2. การเตรียมข้อมูล

- ขั้นตอนที่ 1 import file จาก google drive

```
System
from google.colab import drive
drive.mount('/content/drive')
Mounted at /content/drive
```

- ขั้นตอนที่ 2 ติดตั้ง imbalanced-learn และ pythainlp

```
pip install -U imbalanced-learn
Requirement already satisfied: imbalanced-learn in /usr/local/lib/python3.10/dist-packages (0.10.1)
Collecting imbalanced-learn
  Downloading imbalanced_learn-0.11.0-py3-none-any.whl (235 kB)
    235.6/235.6 kB 3.5 MB/s eta 0:00:00
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn) (1.23.5)
Requirement already satisfied: scipy>=1.5.0 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn) (1.11.3)
Requirement already satisfied: scikit-learn>=1.0.2 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn) (1.2.2)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from imbalanced-learn) (3.2.0)
Installing collected packages: imbalanced-learn
  Attempting uninstall: imbalanced-learn
    Found existing installation: imbalanced-learn 0.10.1
    Uninstalling imbalanced-learn-0.10.1:
      Successfully uninstalled imbalanced-learn-0.10.1
  Successfully installed imbalanced-learn-0.11.0

[ ] pip install pythainlp
Collecting pythainlp
  Downloading pythainlp-4.0.2-py3-none-any.whl (13.4 MB)
    13.4/13.4 MB 72.5 MB/s eta 0:00:00
Requirement already satisfied: requests>=2.22.0 in /usr/local/lib/python3.10/dist-packages (from pythainlp) (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.22.0->pythainlp) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.22.0->pythainlp) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.22.0->pythainlp) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.22.0->pythainlp) (2023.7.22)
Installing collected packages: pythainlp
  Successfully installed pythainlp-4.0.2
```

- ขั้นตอนที่ 3 import libraries ที่ต้องใช้

```
import pandas as pd
import numpy as np
#clean word
import re
from pythainlp.tokenize import word_tokenize
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
from nltk.tokenize.toktok import ToktokTokenizer
#word to vector
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import Pipeline
#model
from sklearn.linear_model import LogisticRegression
#split
from sklearn.model_selection import train_test_split
#Accuracy
from sklearn.metrics import accuracy_score
```

- ขั้นตอนที่ 4 ใช้ไลบรารี pandas เพื่ออ่านข้อมูลจากไฟล์ CSV และตรวจสอบขนาดของ DataFrame ด้วย data.shape แสดงข้อมูลเพิ่มเติมเกี่ยวกับ DataFrame ด้วย data.info()

```
Data preprocessing

[ ] #Load and preprocess the data
data = pd.read_csv('/content/drive/MyDrive/Colab Notebooks/imdb/data/4_imdb_master.csv', encoding='ISO-8859-1') # Replace 'shopping-comment.csv' with your dataset
data.shape
data.info()
```

- ขั้นตอนที่ 5 ใช้เมธอด drop ของ Pandas เพื่อลบคอลัมน์ที่ไม่ต้องการจาก DataFrame data และแสดงข้อมูลและรูปร่างของ DataFrame

```
#Choose the column and remove column
data.drop(['Unnamed: 0'], axis=1, inplace=True)
data.drop(['file'], axis=1, inplace=True)
data.info()
data.head()
data.shape
```

- ขั้นตอนที่ 6 Check dataset
 - ใช้คำสั่ง column เพื่อดูว่า dataset นั้นมี column ไตบ้าง
 - ใช้คำสั่ง value_counts เพื่อนับจำนวนของ data column label
 - ใช้คำสั่ง value_counts เพื่อนับจำนวนของ data column type (train)
 - ใช้คำสั่ง isnull() เพื่อเช็คว่ามีค่าว่างไหม

```
# check dataset
print(data.columns)
print(data['label'].value_counts())
print(data[data['type'] == 'train'].value_counts())
data.isnull().values.any()
```

- ขั้นตอนที่ 7 Clean ข้อความ จะลบอักขระที่ไม่ใช่ตัวอักษร ตัวอักษรเดียว ช่องว่าง และแท็ก HTML

```
# Cleaning the text
def cleaning(sen):
    sen = re.sub('[^A-Za-z]+', ' ', sen)
    sen = re.sub(r'\"\\s+[a-zA-Z]\\s+\", ' ', sen)
    sen = re.sub(r'\"\\s+', ' ', sen)
    sen = re.sub(r'<[^>+>', ' ', sen)
    return sen

data['review'] = data['review'].apply(cleaning)
data.head()
```

- ขั้นตอนที่ 8 ลบ stopwords จะใช้ toktoktokenizer และกำหนดฟังก์ชันเพื่อลบ stopwords ออกจากประโยค

```
#Tokenization of text
tokenizer=ToktokTokenizer()
#Setting English stopwords
stopword_list=nlk.corpus.stopwords.words('english')

#set stopwords to english
stopword=set(stopwords.words('english'))
print(stopword)

#removing the stopwords
def remove_stopwords(text, is_lower_case=False):
    tokens = tokenizer.tokenize(text)
    tokens = [token.strip() for token in tokens]
    if is_lower_case:
        filtered_tokens = [token for token in tokens if token not in stopword_list]
    else:
        filtered_tokens = [token for token in tokens if token.lower() not in stopword_list]
    filtered_text = ' '.join(filtered_tokens)
    return filtered_text

#Apply function on review column
data['review']=data['review'].apply(remove_stopwords)
data.head()

[ ] #variables
X = data['review']
y = data['label']
```

3.ตัวโมเดลที่เลือกใช้

1. Random Forest Model :

Random Forest Model : เป็นหนึ่งในกลุ่มของโมเดลที่เรียกว่า Ensemble learning ที่มีหลักการคือการเทรนโมเดลที่เหมือนกันหลายๆ ครั้ง (หลาย Instance) บนข้อมูลชุดเดียวกัน โดยแต่ละครั้งของการเทรนจะเลือกส่วนของข้อมูลที่เทรนไม่เหมือนกัน แล้วเอาการตัดสินใจของโมเดลเหล่านั้นมาโหวตกันว่า Class ไหนถูกเลือกมากที่สุด โดยขั้นตอนแรกจะนำตัวแปร X และ y มาทำการ train-test-split โดยใช้ test data = 20% และ train data = 80% แบ่งข้อมูลเป็น train กับ test สร้าง TF-IDF vector ซึ่ง max_features ถูกใช้เพื่อกำหนดจำนวนคำสำคัญที่จะถูกใช้ในการสร้าง TF-IDF vector โดยจะเลือกเฉพาะคำที่มีความถี่สูงสุดจำนวน max_features คำ แปลงข้อมูลต้นฉบับในคอลัมน์ review ของข้อมูล dataframe และ data ที่เกี่ยวข้องถูกจัดเก็บในคอลัมน์ label ต่อมาเป็น Random Forest model train กำหนดไว้ n_estimators ซึ่งระบุจำนวนต้นไม้ในโมเดล คือ 100 ซึ่ง class_weight ซึ่งให้น้ำหนักมากกว่ากับคลาสที่มีจำนวนน้อย (balanced) และ random_state ซึ่งกำหนดเพื่อให้ผลลัพธ์สามารถทำซ้ำได้

```
▼ Random Forest Model

[ ] # Training set and Test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Tokenization & Vectorization (ในที่นี้ใช้ TF-IDF)
tfidf_vectorizer = TfidfVectorizer(max_features=2000) # ปรับค่า max_features ตามความเหมาะสม
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)

# Random Forest
rf_model = RandomForestClassifier(n_estimators=100, class_weight='balanced', random_state=42) # ปรับพารามิเตอร์ต่าง ๆ ตามความเหมาะสม
rf_model.fit(X_train_tfidf, y_train)
y_pred = rf_model.predict(X_test_tfidf)

# Accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

Accuracy: 0.50415
```

Link colab

<https://colab.research.google.com/drive/1Q3nyJjwQyYZIZ9IkOmP35rZxRJPwf46F?usp=sharing>

2. Logistic Regression Model : เป็นอัลกอริทึมที่ให้ Output เป็นข้อมูลไม่ต่อเนื่อง เช่น 0 1 (Binary classes) ดังนั้น logistic regression จะใช้สำหรับการจำแนก (Classification) โดยเราจะกำหนดตัวแปรให้กับ TFIDF เพื่อให้ TFIDF ช่วย highlight คำสำคัญของแต่ละประโยค ขั้นตอนต่อไปจะนำตัวแปร X และ y มาทำการ train-test-split โดยใช้ test data = 20% และ train data = 80% ต่อมาคือการ train model จะสร้างตัวแปรให้กับ Logistic regression โดยกำหนดให้ solver='lbfgs' เพื่อป้องกัน warning และเราจะใช้ pipeline ซึ่งเป็น package มาช่วยทำ model ให้สะดวกมากขึ้น เมื่อได้ตัวแปร lr จากการทำ pipeline แล้ว เราก็ใช้คำสั่ง .fit เพื่อ train model ผ่านตัวแปร X_train และ Y_train และทำการ predict ออกมา โดยเราจะวัดค่าความถูกต้องโดยใช้ accuracy_score

```
▼ Logistic Regression Model

# Tokenization & Vectorization (TF-IDF)
tfidf = TfidfVectorizer(lowercase=True)

# Training set and Test set
X_train,X_test,Y_train,Y_test = train_test_split(X,y, test_size=0.20, random_state=225)

# Train Logistic Regression
logicRe =LogisticRegression(solver='lbfgs', max_iter=3000)
lr = Pipeline([('vectorizer',tfidf),('classifier',logicRe)])
lr.fit(X_train,Y_train)
lrpred = lr.predict(X_test)

# Accuracy
print("Accuracy of Logistic Regression => ",accuracy_score(lrpred,Y_test)*100)
```

Link colab

https://colab.research.google.com/drive/1cA4xMuicPFZqXJ_NmmX2XIUNbtv-XKLK?usp=sharing

4.ผลการเปรียบเทียบความถูกต้องของแต่ละโมเดล

— Random Forest Model

Accuracy of Random Forest Model => 0.50415 หรือ 50.415

Random Forest Model

```
[ ] # Training set and Test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Tokenization & Vectorization (ในที่นี้ใช้ TF-IDF)
tfidf_vectorizer = TfidfVectorizer(max_features=2000) # ปรับค่า max_features ตามความเหมาะสม
X_train_tfidf = tfidf_vectorizer.fit_transform(X_train)
X_test_tfidf = tfidf_vectorizer.transform(X_test)

# Random Forest
rf_model = RandomForestClassifier(n_estimators=100, class_weight='balanced', random_state=42) # ปรับพารามิเตอร์ต่าง ๆ ตามความเหมาะสม
rf_model.fit(X_train_tfidf, y_train)
y_pred = rf_model.predict(X_test_tfidf)

# Accuracy
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

Accuracy: 0.50415
```

— Logistic Regression Model

Accuracy of Logistic Regression => 62.975

Logistic Regression Model

```
[20] # Tokenization & Vectorization (TF-IDF)
tfidf = TfidfVectorizer(lowercase=True)

# Training set and Test set
X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size=0.20, random_state=225)

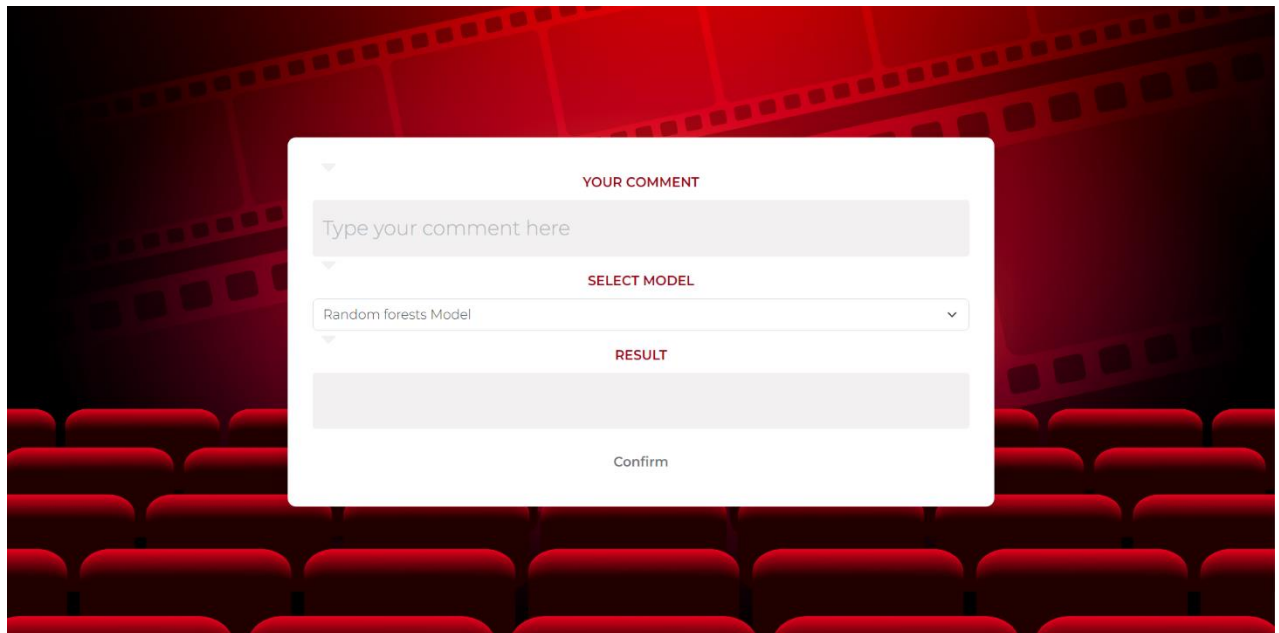
# Train Logistic Regression
logicRe = LogisticRegression(solver='lbfgs', max_iter=3000)
lr = Pipeline([('vectorizer', tfidf), ('classifier', logicRe)])
lr.fit(X_train, Y_train)
lrpred = lr.predict(X_test)

# Accuracy
print("Accuracy of Logistic Regression => ", accuracy_score(lrpred, Y_test)*100)

Accuracy of Logistic Regression => 62.975
```

5.ตัวอย่างหน้าจอการทำงาน

1.หน้าเว็บปกติ สามารถเข้าผ่านลิงก์นี้เพื่อใช้งานได้ <https://project-imdb2-jl2wdacbaa-as.a.run.app>



YOUR COMMENT

Type your comment here

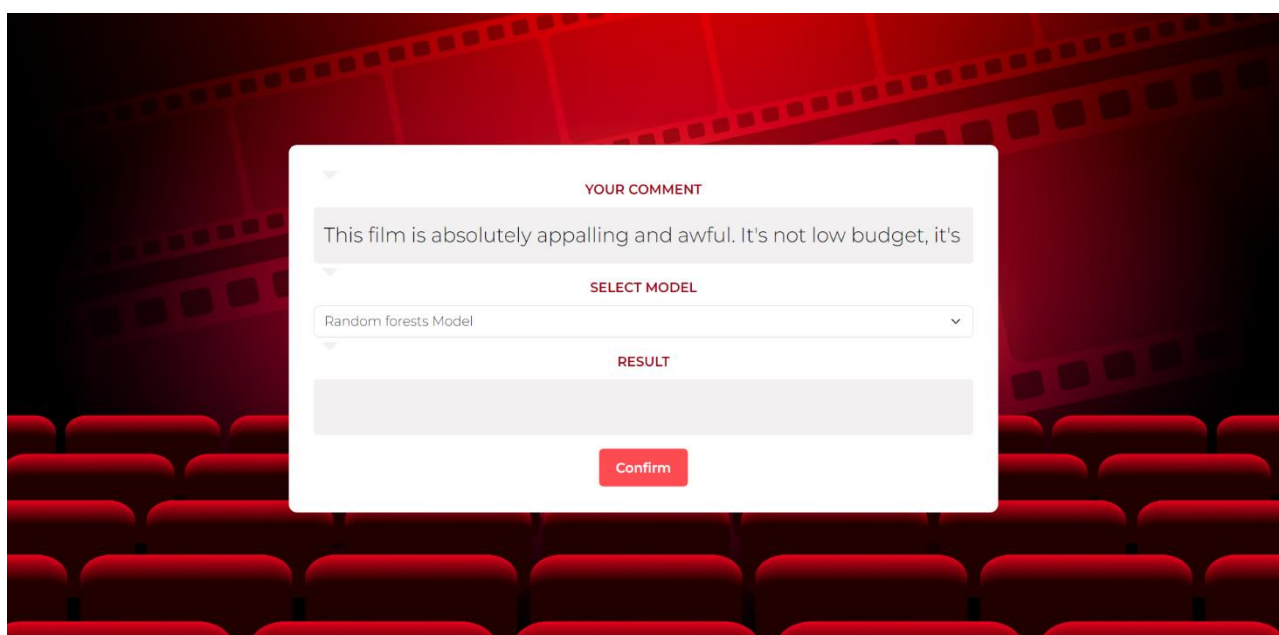
SELECT MODEL

Random forests Model

RESULT

Confirm

2.ป้อน คอมเมนต์ ที่ต้องการอยากจะรู้ว่าเป็นไปในทางที่ดี ที่แย่ หรือ กลางๆ ซึ่งเป็นคอมเมนต์ที่เกี่ยวกับหนังลงไป (ถ้าไม่ได้ใส่ข้อความลงไปช่องก็ไม่สามารถคอมเฟิร์มได้)



YOUR COMMENT

This film is absolutely appalling and awful. It's not low budget, it's

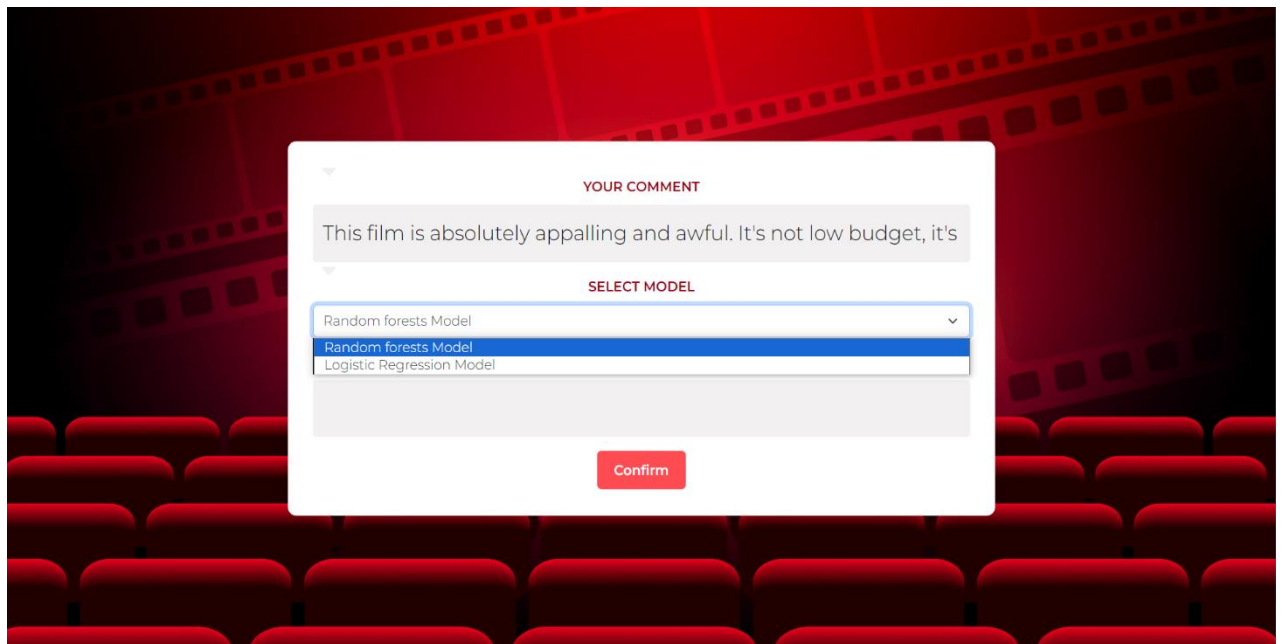
SELECT MODEL

Random forests Model

RESULT

Confirm

3.เลือกโมเดลว่าอยากใช้ตัวไหน มีให้เลือก 2 ตัวเลือก ในที่นี้จะขอใช้ Random forests model
หลังเลือกได้แล้วให้กด Confirm



YOUR COMMENT

This film is absolutely appalling and awful. It's not low budget, it's

SELECT MODEL

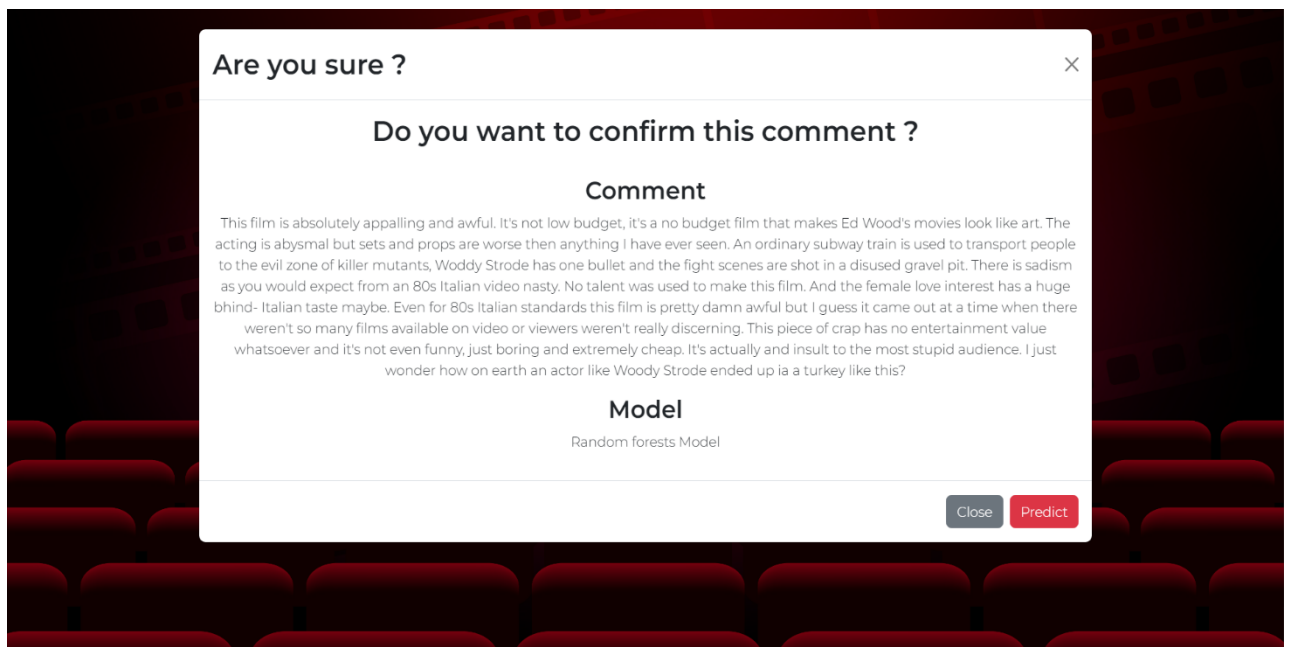
Random forests Model

Random forests Model

Logistic Regression Model

Confirm

4.popup จะขึ้นให้ตรวจสอบว่าถูกต้องไหม ถ้าถูกต้องให้กดได้เลย Predict



Are you sure ?

Do you want to confirm this comment ?

Comment

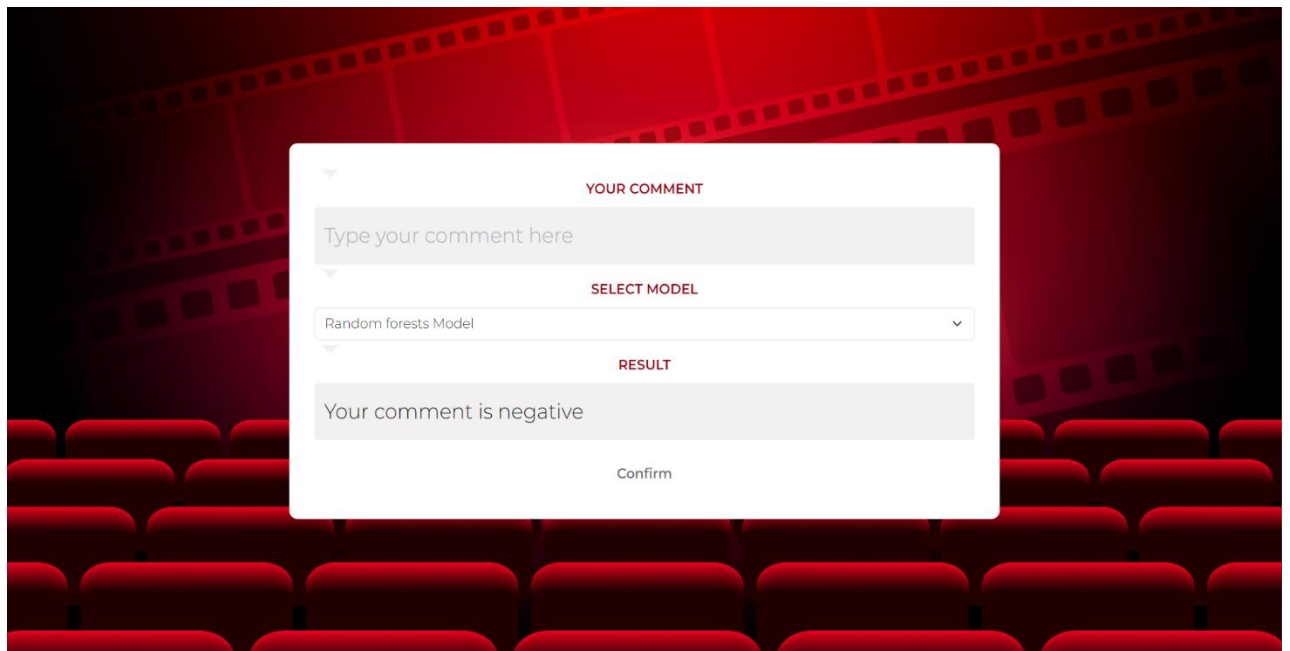
This film is absolutely appalling and awful. It's not low budget, it's a no budget film that makes Ed Wood's movies look like art. The acting is abysmal but sets and props are worse than anything I have ever seen. An ordinary subway train is used to transport people to the evil zone of killer mutants, Woody Strode has one bullet and the fight scenes are shot in a disused gravel pit. There is sadism as you would expect from an 80s Italian video nasty. No talent was used to make this film. And the female love interest has a huge behind- Italian taste maybe. Even for 80s Italian standards this film is pretty damn awful but I guess it came out at a time when there weren't so many films available on video or viewers weren't really discerning. This piece of crap has no entertainment value whatsoever and it's not even funny, just boring and extremely cheap. It's actually an insult to the most stupid audience. I just wonder how on earth an actor like Woody Strode ended up in a turkey like this?

Model

Random forests Model

Close Predict

5.ก็คือหลังจากนำ คอมเมนต์ ไป predict แล้วจะแสดงคำตอบออกมาว่า เป็นคำตอบประเภทไหน positive = pos คือความคิดเห็นเชิงบวก, negative=neg คือความคิดเห็นเชิงลบ และ unsupervised = unsup คือความคิดเห็นกลาง ๆ



YOUR COMMENT

Type your comment here

SELECT MODEL

Random forests Model

RESULT

Your comment is negative

Confirm