ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

# ĐỒ ÁN MÔN HỌC

Máy học

Học kỳ II (2019-2020)

# PHÂN TÍCH CẢM XÚC ĐÁNH GIÁ PHIM


| | |
|---|---|
| Sinh viên 1: | Võ Linh Bảo |
| MSSV: | 18520503 |
| Giảng viên: | Lê Đình Duy |
| | Phạm Nguyễn Trường An |

**Thành phố Hồ Chí Minh, tháng 8 năm 2020**

# PHẦN I MÔ TẢ BÀI TOÁN

## 1 Đặt vấn đề

Nhu cầu giải trí bằng phim ảnh trên thế giới đang chiếm một tỉ trọng đáng kể. Về phía người dùng, họ mong muốn được xem những bộ phim hay, được nhiều người đánh giá cao; và cách đơn giản và tiết kiệm thời gian nhất là dựa trên ý kiến phản hồi từ những người đã xem bộ phim đó. Tuy nhiên, để có được một đánh giá mang tính khách quan nhất chúng ta phải tham khảo từ nhiều ý kiến phản hồi. Để tiết kiệm thời gian cho con người, chúng ta có thể "dạy" cho máy tính đánh giá mức độ cảm xúc (sentiment analysis) của một đánh giá phim bất kỳ, sau đó tổng hợp lại tất cả đánh giá về bộ phim đó, chúng ta sẽ có thể kết luận bộ phim đó có đáng xem (theo ý kiến của phần lớn người xem) hay không.

## 2 Bài toán

Đây là bài toán phân tích cảm xúc tích cực hoặc tiêu cực của một lời đánh giá phim

Input: một đánh giá (dạng văn bản) về bộ phim

Output: đánh giá đó là tích cực (**positive**) hay tiêu cực (**negative**)

# PHẦN II MÔ TẢ DỮ LIỆU

## 1 Cấu trúc bộ dữ liệu

Bộ dữ liệu tự xây dựng gồm 6269 đánh giá phim từ IMDB, bao gồm 4 cột thông tin:

- Tên phim: movie
- Tiêu đề của đánh giá: title
- Nội dung của đánh giá: review
- Nhãn: isPos (0 – positive và 1 – negative)

Khi xây dựng bộ dữ liệu, tôi đã cố gắng để số lượng đánh giá ở 2 nhãn cân bằng nhau: ***positive*** (3144) và ***negative*** (3125).

## 2 Cách thức xây dựng

Sử dụng Selenium để tự động khởi động trình duyệt mở đến trang đánh giá phim, sau đó tải tất cả bình luận (trigger button Load more) rồi tiến hành crawl các đánh giá đó về.

Gán nhãn: tiêu chí gán nhãn được dựa trên rating của bình luận đó:

- Nếu đánh giá có rating 9-10 sao: gán nhãn positive (isPos=0)
- Nếu đánh giá có rating từ 3 sao trở xuống: gán nhãn negative (isPos=1)

## 3 Các thao tác tiền xử lý dữ liệu

- Loại bỏ những ký tự KHÔNG nằm trong 26 chữ cái tiếng Anh, bao gồm: số, dấu chấm câu, ký tự đặc biệt.
- Loại bỏ những từ thừa trong quá trình crawl dữ liệu: “Was this review helpful?Sign into vote.Permalink”
- Chuyển tất cả chữ cái trong đánh giá về chữ thường (lowercase)
- Tokenize đánh giá thành nhiều từ nhỏ (sử dụng thư viện built-in của nltk)
- Loại bỏ stopwords
- Sử dụng kỹ thuật Lemmatization để biến đổi các từ về dạng nguyên mẫu.
- Kết hợp những word token vừa mới xử lý lại thành một câu. Đó là đánh giá ban đầu sau khi qua giai đoạn tiền xử lý.

## PHẦN III RÚT TRÍCH ĐẶC TRƯNG

2 phương pháp được sử dụng để vector hóa các đánh giá thành các vector:

### 1 Count Vectorizer

Khi sử dụng phương pháp này, chúng ta sẽ thu được một ma trận mà trong đó, mỗi hàng sẽ đại diện cho một văn bản, mỗi cột đại diện cho một từ có trong từ điển, và mỗi ô (cell) sẽ chứa tần suất xuất hiện của từ trong văn bản tương ứng.

### 2 Term frequency–inverse document frequency (TF-IDF)

Đầu tiên, TF(Term Frequency) là tần số xuất hiện của 1 từ trong 1 văn bản có cách tính như sau:

$$f_{t,d} \Big/ \sum_{t' \in d} f_{t',d}$$

- **f(t,d)** - số lần xuất hiện từ t trong văn bản d.
- mẫu số là tổng số từ trong văn bản d

Tiếp theo là IDF (**Inverse Document Frequency**): Tần số nghịch của 1 từ trong tập văn bản (corpus).

Mục đích của việc tính IDF là giảm giá trị của các từ thường xuyên xuất hiện như "is", "the"... Do các từ này không mang nhiều ý nghĩa trong việc phân loại văn bản.

$$\mathrm{idf}(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}|}$$

- |D|: tổng số văn bản trong tập D * mẫu số là số văn bản có chứa từ t . Nếu từ đó không xuất hiện ở bất cứ 1 văn bản nào trong tập thì mẫu số sẽ bằng 0, dẫn đến phép chia cho không không hợp lệ, vì thế với trường hợp này thường cộng thêm 1 vào mẫu số.

Và cuối cùng TF-IDF bằng:

$$\mathrm{tfidf}(t, d, D) = \mathrm{tf}(t, d) \times \mathrm{idf}(t, D)$$

Những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều trong văn bản này, và xuất hiện ít trong các văn bản khác. Việc này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao (từ khoá của văn bản đó).

# PHẦN IV CÀI ĐẶT CÁC THUẬT TOÁN MÁY HỌC

Việc cài đặt các thuật toán được thực hiện Google Colab. Ở đây bộ dữ liệu được chia thành 2 tập training và test theo tỉ lệ 80-20. Việc training model được thực hiện bằng cách sử dụng thư viện sklearn và import các thư viện của thuật toán cần training.

## 1 Linear Regression

- Sử dụng TF-IDF

```
Linear Regression (TF-IDF):
              precision    recall  f1-score   support

    Positive       0.92      0.96      0.94       602
    Negative       0.96      0.92      0.94       652

    accuracy                           0.94      1254
   macro avg       0.94      0.94      0.94      1254
weighted avg       0.94      0.94      0.94      1254

Confusion Matrix:
 [[575  27]
 [ 49 603]]
```

- Sử dụng Count Vectorizer

```
Linear Regression (Count Vectorizer):
              precision    recall  f1-score   support

    Positive       0.90      0.94      0.92       602
    Negative       0.94      0.91      0.93       652

    accuracy                           0.92      1254
   macro avg       0.92      0.92      0.92      1254
weighted avg       0.93      0.92      0.92      1254

Confusion Matrix:
 [[567  35]
 [ 60 592]]
```

## 2 Support Vector Machine (SVM)

- Sử dụng TF-IDF

```
SVM (TF-IDF):
              precision    recall  f1-score   support

    Positive       0.92      0.96      0.94       602
    Negative       0.96      0.92      0.94       652

    accuracy                           0.94      1254
   macro avg       0.94      0.94      0.94      1254
weighted avg       0.94      0.94      0.94      1254

Confusion Matrix:
 [[575  27]
 [ 49 603]]
```

- Sử dụng Count Vectorizer

```
SVM (Count Vectorizer):
              precision    recall  f1-score   support

    Positive       0.89      0.92      0.90       602
    Negative       0.92      0.89      0.91       652

    accuracy                           0.91      1254
   macro avg       0.90      0.91      0.91      1254
weighted avg       0.91      0.91      0.91      1254

Confusion Matrix:
 [[552  50]
 [ 69 583]]
```

## 3 Naïve Bayes

- Sử dụng TF-IDF

```
Naive Bayes (TF-IDF):
              precision    recall  f1-score   support

    Positive       0.89      0.97      0.92       602
    Negative       0.97      0.88      0.92       652

    accuracy                           0.92      1254
   macro avg       0.93      0.93      0.92      1254
weighted avg       0.93      0.92      0.92      1254

Confusion Matrix:
 [[582  20]
 [ 75 577]]
```

- Sử dụng Count Vectorizer

```
Naive Bayes (Count Vectorizer):
              precision    recall  f1-score   support

    Positive       0.90      0.97      0.93       602
    Negative       0.97      0.90      0.93       652

    accuracy                           0.93      1254
   macro avg       0.93      0.93      0.93      1254
weighted avg       0.93      0.93      0.93      1254

Confusion Matrix:
 [[582  20]
 [ 67 585]]
```

## PHẦN V ĐÁNH GIÁ, KẾT LUẬN:

- Cả ba thuật toán đều cho kết quả prediction từ **90-94%**
- So sánh với bộ dữ liệu cùng lấy từ IMDB trên Kaggle với số lượng gấp khoảng 8 lần (~50k reviews), một kernel trên Kaggle cũng áp dụng những phương pháp tiền xử lý tương tự và 3 thuật toán nêu trên, kết quả prediction thu được từ **87-89%**
- Nguyên nhân dẫn đến kết quả cao có thể là:
- Số lượng nhãn chỉ là 2
- Trong quá trình thu thập, việc gán nhãn cho dữ liệu được phân định quá rạch ròi (9-10 sao là postive, dưới 3 sao là negative), không có trường hợp nhiễu và trung tính (neutral) (e.g. các đánh giá 5-7 sao)
- Hướng phát triển: kết hợp mở rộng bộ dữ liệu, thêm nhãn (e.g. neutral), trộn từ nhiều nguồn khác nhau