TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

XỬ LÍ NGÔN NGỮ TỰ NHIÊN

NHÓM 15

PHÂN LOẠI VĂN BẢN TỰ ĐỘNG

GIÁO VIÊN HƯỚNG DẪN: PGS.TS LÊ THANH HƯƠNG

SINH VIÊN: NGÔ MINH HẢI
TRẦN THANH TÚ
HOÀNG MẠNH HIỆP

Hà Nội, tháng 10/2019

NỘI DUNG

1. Giới thiệu đề tài
2. Chuẩn bị dữ liệu
3. Các mô hình học máy sử dụng
4. Đã thực hiện

1. Giới thiệu đề tài

- Bài toán phân loại văn bản có thể tự động phân loại một văn bản cho trước để xác định xem văn bản thuộc thể loại nào.

- Bài toán phân loại văn bản là một bài toán học giám sát trong học máy, do nội dung của văn bản đã được gán nhãn, và được sử dụng để thực hiện phân loại.

2. Xử lí dữ liệu

- Dữ liệu lấy ở: https://github.com/duyvuleo/VNTC
- Dữ liệu là các bài báo tiếng Việt, có thể được gán một nhãn trong số 10 nhãn dưới đây:

- Chính trị xã hội
- Đời sống
- Khoa học
- Kinh doanh
- Pháp luật
- Sức khoẻ
- Thế giới
- Thể thao
- Văn hoá
- Vi tính

2.1. Tiền xử lý dữ liệu

- Loại bỏ các kí tự đặc biệt như: dấu chấm, dấu phẩy, dấu mở, đóng ngoặc, … và chuyển chữ hoa về chữ thường sử dụng thư viện gensim.

```
lines = gensim.utils.simple_preprocess(lines)
```

- Sau đó, sử dụng ViTokenizer trong thư viện PyVi để tách từ tiếng Việt.

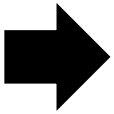
```
lines = ViTokenizer.tokenize(lines)
```

- Cuối cùng, đưa mỗi bài báo về dạng (x,y) trong đó x: bài báo đã xử lí, y: nhãn của bài báo đó.

2.1. Tiền xử lý dữ liệu

- Ví dụ một đoạn văn bản sau khi tiền xử lý:

Ông đã chấp nhận lời mời tham gia lễ kỷ niệm 60 năm ngày D-Day của Tổng thống Pháp Jacque Chirac. Phát ngôn viên của Berlin cho biết: "Tổng thống Chirac đã mời Thủ tướng Schroeder từ trước lễ Giáng sinh và ông đã nhận lời ngay. Thủ tướng cảm thấy rất vui khi được mời". Năm 1994, cố tổng thống Pháp Francois Mitterrand đã không mời cựu thủ tướng Đức Helmut Kohl đến dự lễ kỷ niệm 50 năm sự kiện D-Day.



ông đã chấp_nhận lời mời tham_gia lễ kỷ_niệm năm ngày day của tổng_thống pháp jacque chirac phát_ngôn_viên của berlin cho biết tổng_thống chirac đã mời thủ_tướng schroeder từ trước lễ giáng_sinh và ông đã nhận_lời ngay thủ_tướng cảm_thấy rất vui khi được mời năm cố tổng_thống pháp francois mitterrand đã không mời cựu_thủ_tướng đức helmut kohl đến dự lễ kỷ_niệm năm sự_kiện day

2.2. Feature Engineering

- Để máy tính có thể hiểu được văn bản và phân loại nó thì cần chuyển dữ liệu ở dạng kí tự về dạng số và sử dụng dữ liệu dạng số này để huấn luyện cho máy tính.

- Đưa dữ liệu về dạng vector số học bằng các cách sau:
  - Count vectorizing
  - TF-IDF vectorizing

2.2. Feature Engineering

Count vectorizing

- Khi sử dụng phương pháp này, từ dữ liệu đã tiền xử lý ở trên sẽ thu được một ma trận với:
  - Mỗi hàng đại diện cho một văn bản.
  - Mỗi cột đại diện cho một từ trong từ điển.
  - Mỗi ô sẽ chứa tần suất xuất hiện của từ tương ứng trong văn bản tương ứng.
- Nhược điểm của Count vectorizing: chỉ đếm tần suất xuất hiện của từ trong một văn bản.

2.2. Feature Engineering

TF-IDF vectorizing

- TF-IDF hay Term Frequency - Inverse Document Frequency là một phương pháp vô cùng phổ biến trong xử lý văn bản.
- TF-IDF cũng khắc phục nhược điểm của Count vectorizing.
- Cách xây dựng ma trận tương tự Count vectorizing, riêng giá trị mỗi ô được tính như sau:

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

$$tf(t, d) = \frac{Số\ lần\ xuất\ hiện\ của\ từ\ t\ trong\ d}{Số\ lần\ xuất\ hiện\ nhiều\ nhất\ của\ một\ từ\ trong\ d}$$

$$idf(t, D) = ln\frac{Tổng\ số\ văn\ bản\ trong\ D}{Số\ văn\ bản\ d\ chứa\ từ\ t\ trong\ đó}$$

3. Các mô hình học máy sử dụng

- So sánh hiệu quả của các mô hình học máy khác nhau:
  - Naive Bayes
  - Logistic Regression
  - Random Forest
  - Neuron Network

3.1. Naive Bayes

- Mô hình **Naive Bayes** được xây dựng dựa trên công thức Bayes về xác suất có điều kiện.

- Trong mô hình Naive Bayes, ta giả sử rằng các chiều của dữ liệu $X = \{x_1, x_2, \dots, x_n\}$ độc lập xác suất với nhau. Nghĩa là xác suất để một văn bản có nhãn c được tính như sau:

$$P(X|c) = P(x_1, x_2, \dots, x_n|c) = \prod_{i=1}^{n} P(x_i|c)$$

3.1. Naive Bayes

- Kết quả chạy mô hình Naive Bayes:

```
Validation accuracy:  0.7899881516587678
Test accuracy:  0.7789093363508228
```

- Kết quả chưa thực sự cao.

3.2. Logistic Regression

- Mô hình **Logistic Regression** được xây dựng dựa trên mô hình Linear Regression hay hồi quy tuyết tính hay được áp dụng trong bài toán hồi quy.

- Trong mô hình Logistic Regression, hàm dự đoán thường có dạng:

$$f(x) = \theta(w^T x)$$

- Với $\theta(w^T x)$ thường là hàm sigmoid: $\theta(w^T x) = \frac{1}{1+e^{-w^T x}}$
- Với mô hình này, ta cần xây dựng hàm mất mát.

3.2. Logistic Regression

- Hàm mất mát thường được dùng trong logistic regression:

$$J(w) = -\sum_{i=1}^{N} (y_i \ln(w^T x_i) + y_i \ln(1 - w^T x_i))$$

- Hàm mất mát trên dùng cho bài toán phân loại nhị phân. Để sử dụng được trong bài toán phân loại đa lớp, ta sử dụng chiến lược one vs all.
- Kết quả chạy mô hình Logistic Regression:

```
Validation accuracy:  0.8732227488151659
Test accuracy:  0.8642526750441705
```

3.3. Random Forest

- Mô hình **Random Forest** là một mô hình dạng cây quyết định.
- Random Forest làm việc được với dữ liệu thiếu giá trị.
- Khi Forest có nhiều cây hơn, chúng ta có thể tránh được việc overfitting với tập dữ liệu.

3.3. Random Forest

Các bước sinh Forest:

B1. Chọn ngẫu nhiên “k” features từ tập “m” features (k<<m).

B2. Từ tập “k” features, tính toán ra node “d” là tốt nhất cho node phân loại.

B3. Chia các node con theo node tốt nhất vừa tìm được.

B4. Lặp lại bước 1-3 cho đến khi đạt đến k node.

B5. Lặp lại bước 1-4 để tạo ra “n” cây.

Random Forest Prediction:

B1. Lấy các test features và sử dụng các cây quyết định đã tạo ra để dự đoán kết quả, lưu nó vào một danh sách.

B2. Tính toán số lượng vote trên toàn bộ Forest cho từng kết quả.

B3. Lấy kết quả có số lượng vote lớn nhất làm kết quả cuối cho mô hình.

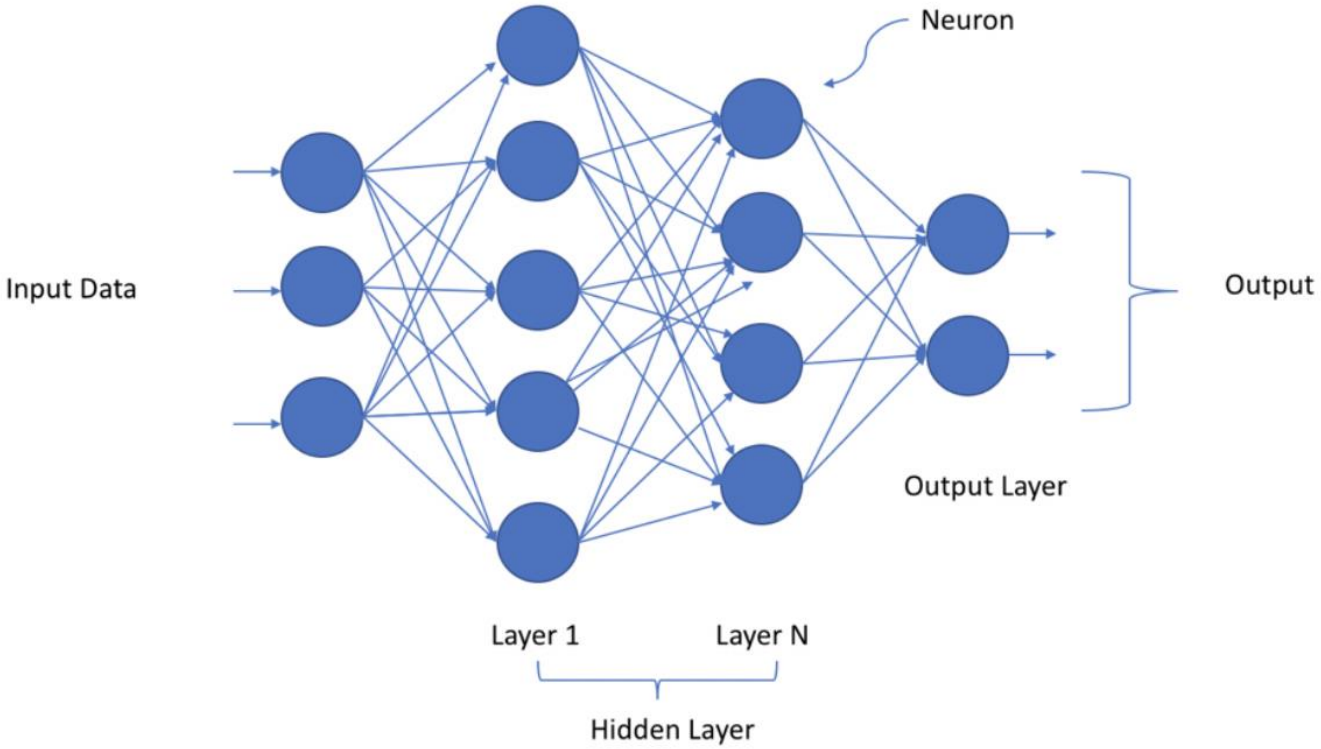
3.3. Random Forest

- Kết quả chạy mô hình Random Forest:

```
Validation accuracy:  0.8545616113744076
Test accuracy:  0.8424949873940405
```

3.4. Deep Neural Network

- **Deep Neural Network** (DNN) là một mô hình mô phỏng lại mạng neuron sinh học.
- Một DNN sẽ gồm nhiều lớp ẩn (Hidden Layer) giữa hai lớp đầu ra và đầu vào.



3.4. Deep Neural Network

- Kết quả chạy DNN:

```
Train on 30383 samples, validate on 3376 samples
Epoch 1/3
30383/30383 [==============================] - 7s 234us/step - loss: 0.9277 -
accuracy: 0.7232 - val_loss: 0.3742 - val_accuracy: 0.8815
Epoch 2/3
30383/30383 [==============================] - 6s 186us/step - loss: 0.3615 -
accuracy: 0.8800 - val_loss: 0.3618 - val_accuracy: 0.8768
Epoch 3/3
30383/30383 [==============================] - 6s 187us/step - loss: 0.3272 -
accuracy: 0.8909 - val_loss: 0.3548 - val_accuracy: 0.8830
Validation accuracy:  0.8829976303317536
Test accuracy:  0.8757072241081532
```

4. Kết luận

- Mô hình Naive Bayes cho kết quả chưa được tốt, một phần vì mô hình khá đơn giản.
- Các mô hình khác cũng chỉ cho kết quả khoảng 85% - 90%, lý do có thể là do:
  - Dữ liệu chưa được sạch, còn nhiều nhiễu.
  - Tiền xử lý chưa thực sự tốt.
  - Các tham số mô hình chưa phù hợp.

Em xin chân thành cảm ơn cô và các bạn đã lắng nghe

Q&A