Report Guides for BDML

1. Tóm tắt

Sự lan truyền của thông tin sai lệch trên các nền tảng truyền thông xã hội là một vấn đề ngày càng gia tăng. Mức độ xác thực của tin tức được đăng trực tuyến không thể được đo lường một cách rõ ràng, vì việc phân loại tin tức theo cách thủ công rất tẻ nhạt và tốn thời gian, đồng thời cũng có thể bị thiên vị. Để giải quyết vấn đề ngày càng gia tăng, các công cụ phát hiện, phân loại và giảm thiểu là nhu cầu hàng ngày. Chúng tôi sử dụng thuật toán decision tree, random forest,Naïve Bayes… với các quy trình Kỹ thuật tính năng và trích xuất trên tập dữ liệu tin tức giả để phân loại một tin bài nhất định. Cuối cùng, chúng tôi sẽ chỉ ra sự khác biệt về kết quả thu thập được, cũng như ứng dụng thực tế của dự án đã thực hiện

1. Giới thiệu

Tin tức giả mạo là một hiện tượng phổ biến trên internet ngày nay. Nó ảnh hưởng đến mọi khía cạnh của cuộc sống của chúng ta. Nó ảnh hưởng đến chúng ta theo những cách khác nhau. Ví dụ, tin tức giả mạo về việc chữa khỏi COVID-19 có thể lan truyền thông tin sai lệch. Để giải quyết thách thức này, bạn được cung cấp một tập dữ liệu để đào tạo mô hình của mình và đưa ra dự đoán cho các tin tức trong tương lai nếu tin tức là thật hay giả.

Input của bài toán là tập các bài báo được thu thập từ nhiều nguồn khác nhau trên phương tiện truyền thông. Ở đây chúng tôi sử dụng thuật toán Decision Tree, Random Forest, Naïve Bayes, Gradient Boost để phân biệt đâu là tin tức đáng tin hay tin giả.

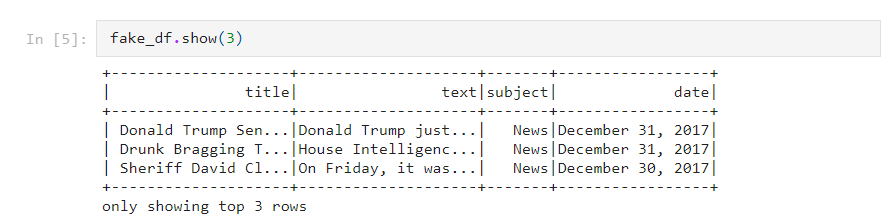
1. Dữ liệu

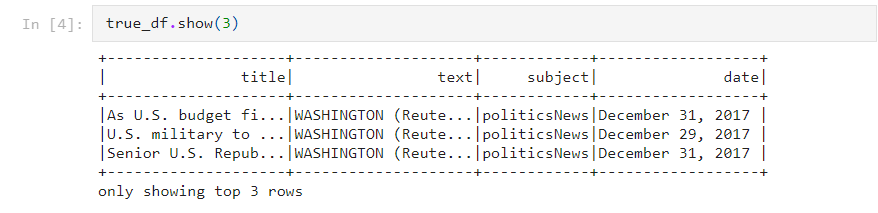
Tập dữ liệu được thu thập từ nhiều nguồn khác nhau; các bài báo trong file true.csv có được bằng cách thu thập thông tin các bài báo từ Reuters.com. Đối với các bài báo giả mạo, chúng được thu thập từ nhiều nguồn khác nhau từ các trang web không đáng tin cậy đã bị Politifact gắn cờ (kiểm tra xác thực tổ chức ở Hoa Kỳ) và Wikipedia. Tập dữ liệu chứa các loại bài báo khác nhau trên các chủ đề, tuy nhiên, phần lớn các bài báo tập trung vào các chủ đề chính trị và tin tức Thế giới.

Dữ liệu bài toán được chia thành 2 file là true.csv và fake.csv

Các cột của tập dữ liệu:

* Title: tiêu đề bài báo
* Text: nội dung của bài báo
* Subject: chủ đề của bài báo
* Date:ngày mà bài báo được phát hành



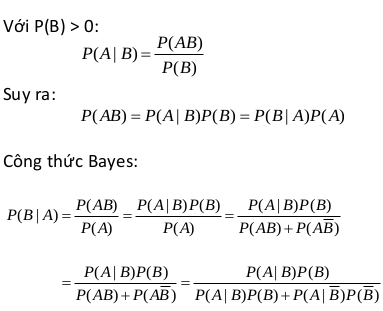


Tiền xử lý dữ liệu:

* Xử lý giá trị null
* Xoá stopword khỏi title,text
* Chuẩn hoá dữ liệu biến text bằng TF-IDF

1. Phương pháp
   1. Naïve Bayes

Công thức Bayes tổng quát:



Thuật toán Naïve Bayes giả định các yếu tố dự đoán độc lập với nhau. Kế đó, Naïve Bayes tính toán xác suất hậu nghiệm P(B|A) từ P(B) và P(A). Trong đó:

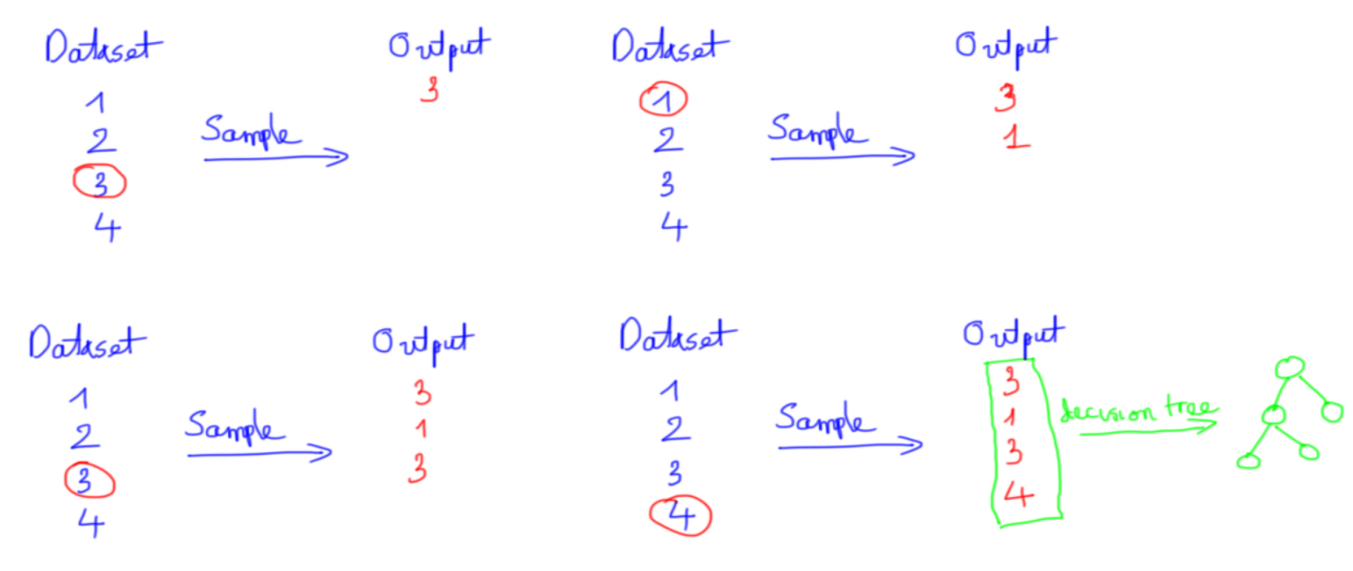
* P(B|A) là xác suất hậu nghiệm của lớp (B, mục tiêu) dự đoán (A, thuộc tính) đã cho.
* P(B): là xác suất tiên nghiệm của lớp.
* P(A|B): là khả năng xảy ra là xác suất của lớp dự đoán đã cho.
* P(A): là xác suất tiên nghiệm của người dự đoán.
  1. Random Forest

Random forest là thuật toán supervised learning, có thể giải quyết cả bài toán regression và classification.

Xây dựng thuật toán Random Forest:

Giả sử bộ dữ liệu có n dữ liệu (sample) và mỗi dữ liệu có d thuộc tính (feature).

Lấy ngẫu nhiên n dữ liệu từ bộ dữ liệu với kĩ thuật [Bootstrapping](https://en.wikipedia.org/wiki/Bootstrapping_(statistics)) (**random sampling with replacement)**. Tức khi sample được 1 dữ liệu thì không bỏ dữ liệu đấy ra mà vẫn giữ lại trong tập dữ liệu ban đầu, rồi tiếp tục sample cho tới khi sample đủ n dữ liệu. Khi dùng kĩ thuật này thì tập n dữ liệu mới có thể có những dữ liệu bị trùng nhau.



Sau khi sample được n dữ liệu từ bước 1 thì chọn ngẫu nhiên ở k thuộc tính (k < n). Sau đó sẽ thu được bộ dữ liệu mới gồm n dữ liệu và mỗi dữ liệu có k thuộc tính.

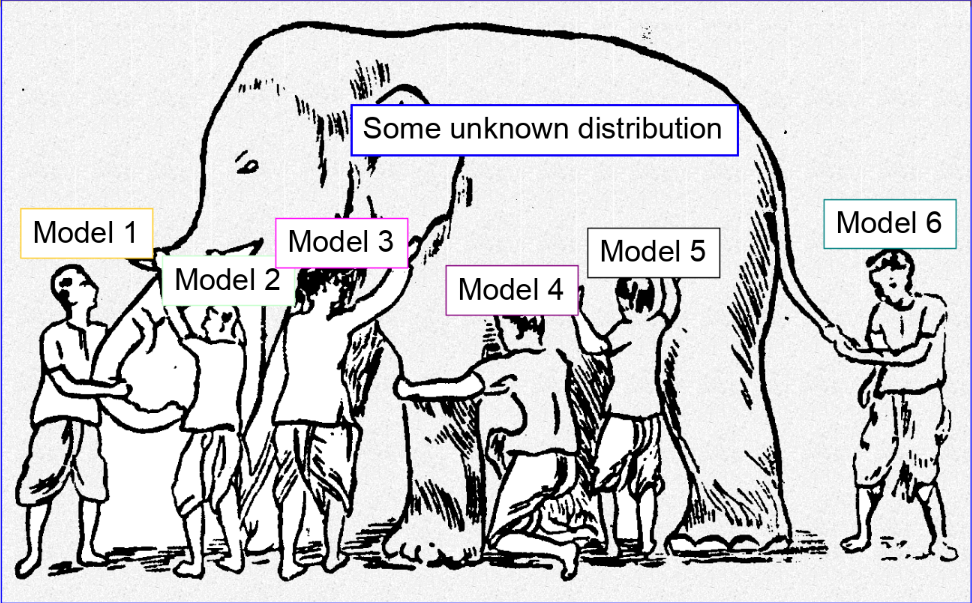
Dùng thuật toán Decision Tree để xây dựng cây quyết định với bộ dữ liệu ở bước 2.

Do quá trính xây dựng mỗi cây quyết định đều có yếu tố ngẫu nhiên (random) nên kết quả là các cây quyết định trong thuật toán Random Forest có thể khác nhau.

Thuật toán Random Forest sẽ bao gồm nhiều cây quyết định, mỗi cây được xây dựng dùng thuật toán Decision Tree trên tập dữ liệu khác nhau và dùng tập thuộc tính khác nhau. Sau đó kết quả dự đoán của thuật toán Random Forest sẽ được tổng hợp từ các cây quyết định.

Khi dùng thuật toán Random Forest, có các thuộc tính cần chú ý: số lượng cây quyết định sẽ xây dựng, số lượng thuộc tính dùng để xây dựng cây. Ngoài ra, vẫn có các thuộc tính của thuật toán Decision Tree để xây dựng cây như độ sâu tối đa, số phần tử tối thiểu trong 1 node để có thể tách.

* 1. Gradient Boost



Vấn đề tối ưu của Gradient Boosting: min*cn*​,*wn*​​*L*(*y*,*Wn*−1​+*cn*​*wn*​))

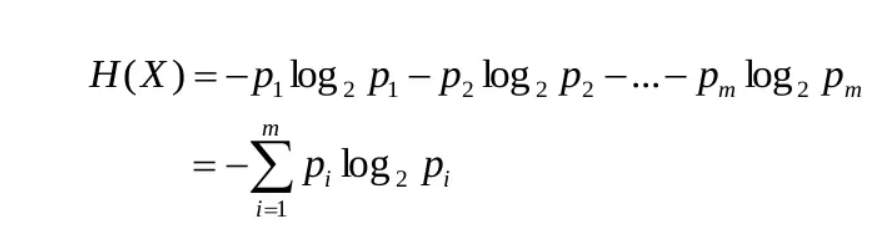
Nếu chúng ta coi chuỗi các model boosting là một hàm số W, thì mỗi hàm learner có thể coi là một tham số w. Để cực tiểu hóa hàm loss L(y, W), chúng ta áp dụng Gradient Descent: *Wn*​ = *Wn*−1 ​– *η(*∂/*w*∂)​*L*(*Wn*−1​)

Đến đây, ta có thể thấy mối quan hệ liên quan sau: *cn*​*wn*​ ≈ −*η(*∂/*w*∂)​*L*(*Wn*−1​)

Với *wn*​ là model được thêm vào tiếp theo. Khi đó, model mới cần học để fit để vào giá trị −*η(*∂/*w*∂)​*L*(*Wn*−1​) [\*]. (Giá trị [\*] còn có 1 tên gọi khác là **pseudo-residuals**)

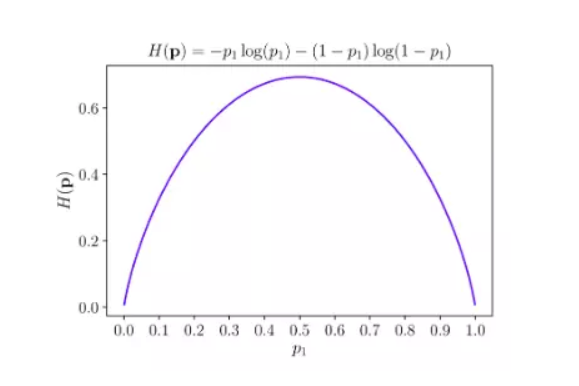
Tóm lại, chúng ta có thể tóm tắt quá trình triển khai thuật toán như sau:

* Khởi tạo giá trị pseudo-residuals là bằng nhau cho từng điểm dữ liệu
* Tại vòng lặp thứ i
  + Train model mới được thêm vào để fit vào giá trị của pseudo-residuals đã có
  + Tính toán giá trị confidence score *ci*​ của model vừa train
  + Cập nhật model chính *W*=*W*+*ci*​∗*wi*
  + Cuối cùng, tính toán giá trị pseudo-residuals – *η(*∂/*w*∂)​*L*(*Wn*−1​) để làm label cho model tiếp theo
* Sau đó lặp lại với vòng lặp i+1.
  1. Desicion tree
* Decision tree là một mô hình supervised learning, có thể được áp dụng vào cả hai bài toán classification và regression.
* Xây dựng mô hình bằng cách chai mẫu theo giá trị của biến số
* Khi mà dự đoán nó sẽ đi qua từng nhánh cho tới nhánh cuối và sẽ dự đoán tại nhánh lá
* Giả sử ta có biến ngẫu nhiên rời rạc X



H(X) là entropy của X (0 <= H(X) <=

Nếu ta xét m = 2, tức là pi = 0 và giá trị kia bằng 1. Khi đó H(X) = 0



Tóm lại thì

* Mô hình này dễ sử dụng
* Cần ít dữ liệu
* Xây dựng nhanh và phân lớp nhanh

Tuy nhiên

* Không đảm bảo việc tối ưu trên cây
* Có thể bị overfiting

1. Thực nghiệm, kết quả, thảo luận
   1. Thực nghiệm

Nhóm sử dụng Cross Validation để điều chỉnh các siêu tham số của các mô hình để chọn mô hình tốt nhất. Dữ liệu được chia làm 10 phần. Độ đo sử dụng cho bài toán phân loại này gồm accuracy và confusion matrix.

**Accuracy** là độ đo sự tương đồng giữa kết quả dự đoán và kết quả. Accuracy càng cao thì phép đo đó rất gần với giá trị đích.

**Accuracy=(TP+TN)/(TP+TN+FP+FN)**

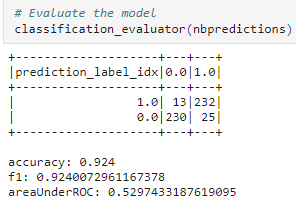
**F1** là số dung hòa Recall và Precision giúp ta có căn cứ để lựa chọn model. F1 càng cao càng tốt.

**F1**=2\*precion\*recall/(precision+recall)

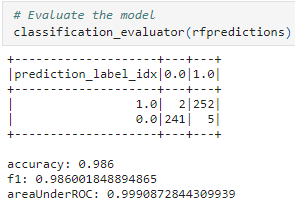
**Area Under the ROC**: Là vùng nằm dưới ROC, vùng này càng lớn thì model càng tốt.

**Confusion matrix** gồm các giá trị sau: P, N, TP, TN, FP, FN. Giả sử đây là bài toán phân loại, biến mục tiêu của tập dữ liệu có 2 loại nhãn là dương tính và âm tính, ta có thể hiểu các thuật giá trị trong confusion matrix như sau

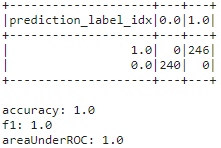
* P (Condition positive) là tổng số trường hợp dương tính thực tế.
* N (Condition Negative) là tổng số trường hợp âm tính thực tế.
* TP (True positive) là số các trường hợp dự đoán dương tính.
* TN (True negative) là số các trường hợp dự đoán âm tính đúng.
* FP (False positive) là số các trường hợp dự đoán dương tính sai.
* FN (False negative) là số các trường hợp dự đoán âm tính sai.
  1. Kết quả
* *Kết quả của thuật toán Naïve Bayes:*



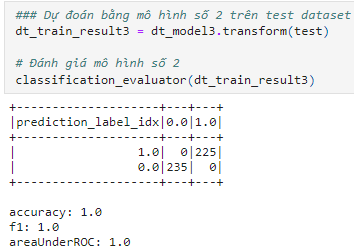
* *Kết quả của thuật toán Random Forest*



* *Kết quả của thuật toán Gradient Boosting*



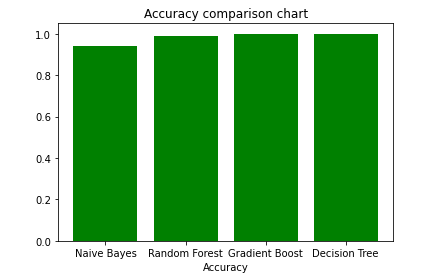
* *Kết quả của thuật toán Decision Tree*



* 1. Thảo luận

Sau khi qua thực nghiệm chúng em đã trực tiếp sử dụng 1 bài báo để đánh giá xem thuật toán nào cho kết quả đúng nhất. Thật may là cả 4 thuật toán đều cho kết quả là fake đúng với bài báo đăng sai mà chúng em thực nghiệm.

1. Kết luận



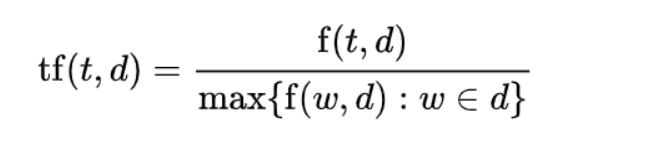
Trong 3 thuật toán, kết quả dự đoán của Gradient Boost và Decision Tree tốt nhất. Random Forest cũng khá tốt, nhưng dùng trên tập dữ liệu lớn lại quá tốn thời gian. Naive Bayse ban đầu được dự đoán là thuật toán phù hợp nhất với bài toán này, nhưng từ kết quả, có thể thấy kết quả dự đoán Naive Bayes là không bằng 3 thuật toán còn lại. Bù lại, Naive Bayes là thuật toán nhanh nhất trong 4 thuật toán, dù tăng kích cỡ tập dữ liệu lên 4 lần vẫn nhanh hơn 3 thuật toán còn lại.

1. Phụ lục

Văn bản, hay ngôn ngữ tự nhiên, là một dạng rất khó xử lý đơn giản vì các đặc điểm và phong cách ngôn ngữ khác nhau như châm biếm, ẩn dụ, v.v. Hơn nữa, có hàng nghìn ngôn ngữ nói và mỗi ngôn ngữ đều có ngữ pháp, chữ viết và cú pháp riêng. Xử lý ngôn ngữ tự nhiên là một nhánh của trí tuệ nhân tạo và nó bao gồm các kỹ thuật có thể sử dụng văn bản, tạo mô hình và đưa ra dự đoán. Ở đây nhóm sử dụng kỉ thuật TF-IDF để đánh giá tầm quan trọng của một từ trong văn bản. Mục đích của công việc này là tạo ra một hệ thống hoặc mô hình có thể sử dụng dữ liệu của các bản tin trong quá khứ và dự đoán khả năng một bản tin có phải là giả mạo hay không.

- TF-**Term frequency (tần suất xuất hiện của từ):**

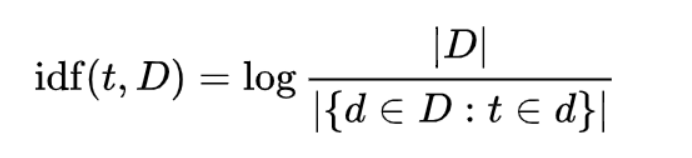
* là số lần từ xuất hiện trong văn bản
* một thuật ngữ xuất hiện càng nhiều , thường xuyên thì thuật ngữ đó càng có khả năng mang lại thông tin hữu ích cho văn bản(trừ stop-word)



* tf(t, d): tần suất xuất hiện của từ t trong văn bản d
* f(t, d): Số lần xuất hiện của từ t trong văn bản d
* max({f(w, d) : w ∈ d}): Số lần xuất hiện của từ có số lần xuất hiện nhiều nhất trong văn bản d

- IDF-**Inverse document frequency (nghịch đảo tần suất của văn bản):**

* giúp đánh giá tầm quan trọng của 1 từ
* nếu một thuật ngữ xuất hiện hầu hết trong văn bản nó sẽ có rất ít khả năg hữu ích để phân biệt và phân loại chúng



* idf(t, D): giá trị idf của từ t trong tập văn bản
* |D|: Tổng số văn bản trong tập D
* |{d ∈ D : t ∈ d}|: thể hiện số văn bản trong tập D có chứa từ t.

1. Đóng góp

|  |  |  |
| --- | --- | --- |
| STT | Thành viên | Công việc |
| 1 | Trần Gia Nguyên | Giải quyết bài toán bằng thuật toán Naïve Bayes.  Tổng kết, so sánh các thuật toán. Tiền xử lý dữ liệu |
| 2 | Huỳnh Thị Hương Ly | Giải quyết bài toán bằng thuật toán Gradient Boost  Tiền xử lý dữ liệu |
| 3 | Phan Thành Trung | Giải quyết bài toán bằng thuật toán Random Forest |
| 4 | Huỳnh Thiên Long | Giải quyết bài toán bằng thuật toán Decision Tree |

1. Tham khảo
2. Ahmed H, Traore I, Saad S. (2017) “Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In: Traore I., Woungang I., Awad A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science, vol 10618. Springer, Cham (pp. 127-138).
3. Ahmed H, Traore I, Saad S. “Detecting opinion spams and fake news using text classification”, Journal of Security and Privacy, Volume 1, Issue 1, Wiley, January/February 2018.
4. http://www.ijirset.com/upload/2020/june/115\_4\_Source.PDF?fbclid=IwAR0HeJY2lqy-3YUXulldTykU1NW2D4AI-ds8v3tzTbxUYs0fhIgE0d9OSDY