

## โปรแกรมตรวจจับข่าวปลอมด้วย RandomForest

### ที่มาและความสำคัญ

ประเด็นข่าวปลอมกำลังเป็นที่น่าสนใจในสังคมปัจจุบัน ขณะนี้อินเทอร์เน็ตทำให้เกิดผู้ผลิตข่าวสารขึ้นมากมายโดยไม่จำเป็นต้องเป็นสำนักข่าวใหญ่เหมือนในสมัยก่อนที่ผู้คนรับรู้ข่าวสารทางการประกาศข่าวจากโทรทัศน์ วิทยุ หรือ หนังสือพิมพ์ ดังนั้นการคัดกรองข่าวสารจึงกลายเป็นความลำบากของผู้รับสารที่จะต้องใช้วิจารณญาณในการรับข้อมูล นอกจากนี้ ข่าวกรองยังสามารถสร้างความเสียหายให้กับบุคคล ๆ หนึ่ง และสังคมโดยกว้าง เช่น กรณีข่าวปลอมในการเลือกตั้งของประเทศสหรัฐอเมริกาที่มีข่าวปลอมออกมาทำให้ผู้สมัครเสียความน่าเชื่อถือ และอาจมีส่วนทำให้เสียคะแนนผลการเลือกตั้ง

ทั้งนี้ เนื่องจากปริมาณข่าวปลอมมีจำนวนมาก การเรียนรู้ด้วยเครื่อง(Machine Learning) จึงเป็นเครื่องมือที่น่าสนใจที่จะนำมาใช้แทนกำลังคน และช่วยประหยัดเวลา โดยโปรแกรมนี้จะใช้โมเดล RandomForest จากแพ็คเกจของ scikit learn เพื่อทำ Supervised learning โดยการจำแนกกลุ่ม (Classification) ระหว่างข่าวจริงและข่าวปลอม

### วิธีการดำเนินงาน

1. หาชุดข้อมูลของข่าวจริงและข่าวปลอมที่มีการติดป้าย (label) จำแนกไว้แล้ว จาก <https://www.kaggle.com/jruvika/fake-news-detection>
  - a. ข้อมูลประกอบด้วย URL, หัวข่าว, เนื้อข่าว, label
  - b. ข้อมูลจัดเก็บอยู่ในรูปของไฟล์ .csv
2. ศึกษางานวิจัยที่เกี่ยวข้อง เพื่อหา feature ที่เหมาะสมสำหรับการตรวจจับข่าวปลอม
  - a. จากงานวิจัย **Automatic Detection of Fake News** ของ Veronica Perez-Rosas et al.<sup>1</sup> กล่าวว่า feature ที่ใช้มี 4 กลุ่ม ได้แก่
    - Punctuation (period, comma, dash, question mark, and exclamation mark)
    - Psycholinguistic features จาก LIWC lexicons (Style, tone, perception words, etc.)
    - Readability (the number of characters, complex words, long words, number of syllables, word types, and number of paragraphs)
    - Syntax

หมายเหตุ ทั้งนี้ นิสิตเลือก Punctuation และ Readability มาประยุกต์กับโปรแกรมนี้นี้เท่านั้น
3. ศึกษาวิธีการปรับ parameter ของ RandomForest เพื่อหา parameter ที่ดีที่สุดที่ทำให้โมเดลทำงานได้ดีที่สุดในเวลาที่จำกัด

---

<sup>1</sup> Perez-Rosas et al. "Automatic Detection of Fake News." *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA, Aug. 20-26, 2018, pp. 3391-3401. <http://aclweb.org/anthology/C18-1287>.

- a. จากบทความของ William Koehrsen<sup>2</sup> เสนอการใช้ Random Search Cross Validation เพื่อหา parameter ที่เหมาะสม

หมายเหตุ Code สำหรับหา parameter ที่เหมาะสม (ดังภาพด้านล่าง) ไม่ได้อยู่ในไฟล์

fake\_news\_detection.py Code ส่วนนี้ เขียนขึ้นมาเพื่อหา parameter สำหรับฝึกโมเดลเท่านั้น

```
# This is codes of William Koehrsen for hyperparameter tuning.
# resource: https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python
# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 200, stop = 2000, num = 10)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(10, 110, num = 11)]
max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 4]
# Method of selecting samples for training each tree
bootstrap = [True, False]
# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf,
               'bootstrap': bootstrap}
rf_random = RandomizedSearchCV(estimator = self.model, param_distributions = random_grid,
                               n_iter = 100, cv = 3, verbose=2, random_state=42,
                               n_jobs = -1)
rf_random.fit(train_sparse_feature_matrix, train_label_list)
print(rf_random.best_params_)
```

#### 4. แพ็กเกจที่ใช้

- ใช้ RandomForestClassifier จากแพ็กเกจ sklearn.ensemble ในการฝึกโมเดลและทำนายผล
- ใช้แพ็กเกจ pandas ในการเปิดไฟล์ csv เพื่อจะได้เข้าถึงข้อมูลของแต่ละหมวดได้ด้วย key ใน dictionary ซึ่ง key คือหัวตาราง
- ใช้ precision\_recall\_fscore\_support จากแพ็กเกจ sklearn.metrics และ method .score(X,Y) ของ sklearn.ensemble.RandomForestClassifier ในการประเมินผลความสามารถของโมเดล
- ใช้คำสั่ง .punctuation จากแพ็กเกจ string ในระบุ punctuation

<sup>2</sup> Koehrsen, William. (2018). "Hyperparameter Tuning the Random Forest in Python" [Online]. Retrieve Dec 12, 2018 from <https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74>.

- e. ใช้แพ็คเกจ re ในการเขียน regular expression ดึงแหล่งข้อมูลของข่าวมาเก็บไว้โดยตัดรายละเอียดของเว็บออก ซึ่ง regular expression ของ url ได้มาจาก <https://www.sitepoint.com/community/t/website-regex-pattern/274741>
  - f. ใช้คำสั่ง word\_tokenize จากแพ็คเกจ nltk.tokenize เพื่อแบ่งคำ
5. Code ที่เขียนขึ้น
- a. Class Data\_processing() สำหรับการเตรียมข้อมูล และการสกัด feature
    - i. ฟังก์ชัน tokenizer() สำหรับการแบ่งคำ แล้วแยกเป็น 2 หมวด ได้แก่ lexical word กับ punctuation เพื่อแยกเก็บคนละ feature
    - ii. ฟังก์ชัน Lexical\_diversity() สำหรับคำนวณหาความหลากหลายของการใช้คำ เพื่อเก็บเป็น feature
    - iii. ฟังก์ชัน bigram\_generator() สำหรับสร้าง bigram เพื่อเก็บเป็น feature
    - iv. ฟังก์ชัน get\_feature() สำหรับการสกัด feature ต่าง ๆ ออกมาจากข้อมูล ได้แก่ คำ, ความยาวของตัวอักษร, จำนวนย่อหน้าในเนื้อข่าว, ความหลากหลายของการใช้คำในหัวข่าวก้าว, ความหลากหลายของการใช้คำในเนื้อข่าว, สำนักรข่าวจาก url และ bigram
  - b. Subclass DetectFakeNews()
    - i. ฟังก์ชัน train() สำหรับฝึกโมเดล
    - ii. ฟังก์ชัน detect() สำหรับตรวจสอบข่าวโดยมีพารามิเตอร์ 3 ตัว ได้แก่ ที่อยู่เว็บไซต์ หัวข่าวและเนื้อข่าว
    - iii. ฟังก์ชัน evaluation() สำหรับตรวจสอบความสามารถของโมเดล

## ผลการดำเนินงาน

โมเดลสามารถตรวจสอบข่าวปลอมได้โดยการเรียกฟังก์ชัน detect() แล้วใส่พารามิเตอร์ที่เป็น string จำนวน 3 ตัว ได้แก่ ที่อยู่เว็บไซต์ หัวข่าว และเนื้อข่าว โดยระยะเวลาที่ใช้ในการฝึกจะใช้เวลาประมาณ 4 นาที และมีระยะเวลาที่ใช้ในการตรวจสอบข่าวปลอมประมาณ 30 วินาที

จากการทดสอบโดยการเรียกฟังก์ชัน detect() แล้วใส่ข้อมูลข่าวในปัจจุบัน โมเดลสามารถทำนายว่าข่าวใดเป็นข่าวจริงหรือข่าวปลอมได้อย่างแม่นยำ อีกทั้ง เมื่อทดสอบความสามารถของโมเดลโดยใช้ฟังก์ชัน evaluation() กับข้อมูลที่แยกไว้สำหรับการทดสอบได้ค่าต่าง ๆ ดังนี้

Accuracy	0.9786432160804021
Precision	0.9795765459689192
Recall	0.9786432160804021
f1	0.9786633101146817

ทั้งนี้ แม้ว่าจะใช้ข่าวในปัจจุบันในการทดสอบ แต่ข้อมูลเกี่ยวกับสำนักข่าวปลอมที่ใช้ทดสอบก็ยังมาจากสำนักข่าวเดียวกับข้อมูลที่ใช้ฝึก จึงอาจทำให้โมเดลเกิดการ overfitting ได้ ถ้าสามารถหาข่าวจากสำนักข่าวที่ผลิตข่าวปลอมใหม่ได้ อาจจะสามารถทดสอบความสามารถของโมเดลได้ดียิ่งขึ้น

### อุปสรรคที่พบ

การหา feature ที่ใช้ในการฝึกโมเดลทำได้ค่อนข้างยาก เพราะ method ของ `RandomForestClassifier.feature_importances_` ไม่แสดงค่าที่ใช้ในโมเดลออกมาเป็นตัวค่า นิสิตพยายามหา Code ที่จะสามารถนำมาช่วยแสดงแล้ว แต่ Code หลาย ๆ แพล่งต้องใช้ความเข้าใจทางสถิติร่วมด้วย หากมีเวลามากกว่านี้ในการทำ ความเข้าใจเรื่องสถิติ อาจทำให้หา feature ที่เหมาะสมได้ดียิ่งขึ้น

นอกจากนี้นิสิตพยายามจะทำให้ฟังก์ชัน `detect()` มีลักษณะแบบ interactive เพื่อลดปัญหาการใส่ data structure ของพารามิเตอร์ผิด โดยการสร้างฟังก์ชันใหม่ที่รับพารามิเตอร์เข้ามาโดยใช้คำสั่ง `input()` แต่พบปัญหาตรงส่วนของเนื้อหาข่าวที่ไม่สามารถนำข้อมูลเข้าไปในฟังก์ชันได้ อาจเป็นเพราะขนาดที่ใหญ่เกินไป หรือการเพิ่มบรรทัดจากการขึ้นย่อหน้าใหม่ในเนื้อหาข่าว

### สรุปผล

จากการทำโปรเจกต์ได้ประยุกต์ใช้การเขียนโปรแกรมด้วยภาษาไพธอนค่อนข้างมาก และทำให้เข้าใจกระบวนการทำ supervised learning และการประหยัด Code ด้วยการใช้ class เพิ่มขึ้นเมื่อนิสิตต้องวางแผนโปรเจกต์ทั้งหมดด้วยตัวเอง นอกจากนี้ยังได้เรียนรู้เกี่ยวกับ machine learning ประเภท ensemble การประยุกต์ใช้แพ็คเกจต่าง ๆ งานวิจัยทางด้านการประมวลผลภาษาธรรมชาติเพิ่มมากขึ้น ซึ่งสิ่งเหล่านี้ช่วยให้นิสิตได้เปิดโลกทัศน์ทางศาสตร์นี้ให้ชัดเจนและกว้างขึ้น