DS4Biz - *Assignment 2* (25 Marks)

Text Scraping & Classification

Deadline: วันศุกร์ที่ 27 พฤศจิกายน 2563 เวลา 24.00 น.

Submission: *Two*-People Group github repository – stage, commit with message, and push

Overview

จุดประสงค์ของ Assignment นี้ คือ ให้นศ. ทำการ scrape (รวบรวม) ข้อมูลจากหน้าเว็บไซต์ จำนวน 2 เว็บ ได้แก่

1. เว็บจำลองเก็บรวบรวมข่าวเก่า
   http://www.it.kmitl.ac.th/~teerapong/news_archive/index.html
   ให้นศ. ทำการ scrape บทความข่าวจากหน้าเว็บไซต์ (มีจำนวนมากกว่า 1 หน้าเว็บ) **ทั้งหมดทุกบทความ** โดยมีเป้าหมายเพื่อพัฒนาระบบที่มี Machine Learning/Data Science ในการช่วยบรรณาธิการข่าว ในการจัดหมวดหมู่ข่าวอัตโนมัติ เช่น Technology, Business และ Sport โดยปัญหานี้ให้นศ.มองเป็น **Multi-class classification problem** โดยข่าว 1 บทความ สามารถอยู่ได้ในหมวดหมู่เดียวจาก 3 classes โดยขณะที่นศ. ทำการ scrape ข้อมูล ให้ทำการเก็บหมวดหมู่ข่าวมาเป็นตัวอย่างในการ train/fit model ของนศ. โดยนศ. สามารถทดลองใช้เฉพาะเนื้อข่าว (body/content) หรือหัวข้อข่าว (heading/title) หรือรวมกันก็ได้ เพื่อให้ได้ประสิทธิภาพสูงสุด (ความแม่นยำ (accuracy) ปัจจุบันอยู่ที่ ~0.975) รวมถึงขั้นตอนการทำ Text Processing อยู่ที่นศ.จะเลือกทดลองใช้ และทำการตั้งค่าการทดลอง (Experimental Settings) อย่างถูกต้องตามหลักวิชาการ (เช่น การแบ่ง train/test dataset การใช้ hold-out sampling หรือ cross validation ตามที่นศ. เห็นเหมาะสม และต้องให้เหตุผลการเลือกใช้) ให้ได้ชนิดของโมเดลและพารามิเตอร์ที่ทำให้โมเดลมีความแม่นยำที่สุด

News Article Archive

Archive of all news headlines and stories, organised per month.

Articles — January [118]
Articles — February [124]
Articles — March [116]
Articles — April [118]
Articles — May [115]
Articles — June [115]
Articles — July [122]
Articles — August [116]
Articles — September [113]
Articles — October [124]
Articles — November [122]
Articles — December [115]

2. เว็บฝึกการ scrape ข้อมูล ซึ่งแสดงคำกล่าว (Quote) ของผู้มีชื่อเสียง

   https://quotes.toscrape.com/

   ให้นศ. ทำการ scrape “quote” จากหน้าเว็บไซต์ (มีจำนวนมากกว่า 1 หน้าเว็บ) **ทั้งหมดทุก ๆ quote** โดยมีเป้าหมายเพื่อพัฒนาระบบที่มี Machine Learning/Data Science ในการช่วยติด Tag แก่ quote อย่างอัตโนมัติ เช่น Love, Inspirational, Abilities, Deep-thoughts เป็นต้น โดยปัญหานี้ให้นศ.มองเป็น **Binary classification problem** ว่า quote นี้ควรทำการติด tag เช่น “humor” หรือไม่ (Yes/No)

   โดย 1 quote สามารถติด tag ได้มากกว่าหรือเท่ากับ 1 tag โดยขณะที่นศ.ทำการ scrape ข้อมูลให้ทำการเก็บ tags มาเป็นตัวอย่างในการ train/fit model ของนศ. โดยนศ.สามารถทดลองใช้เฉพาะเนื้อคำของ quote (body/content) หรือชื่อผู้แต่ง (author) หรือรายละเอียดของผู้แต่ง (about author/author detail) หรือนำมาร่วมกันก็ได้ เพื่อให้ได้ประสิทธิภาพสูงสุด รวมถึงขั้นตอนการทำ Text Processing อยู่ที่นศ. จะเลือกทดลองใช้และทำการตั้งค่าการทดลอง (Experimental Settings) อย่างถูกต้องตามหลักวิชาการ (เช่น การแบ่ง train/test dataset การใช้ hold-out sampling หรือ cross validation ตามที่นศ.เห็นเหมาะสม และต้องให้เหตุผลการเลือกใช้) ให้ได้ชนิดของโมเดลและพารามิเตอร์ที่ทำให้โมเดลมีความแม่นยำที่สุด

สำหรับ Assignment นี้ นศ.จะต้องถูกทำลงใน Jupyter Notebook โดยแยกการทำการทดลองของนศ. เป็น 2 Notebook โดยแต่ละ Notebook ใช้กับคนละเว็บกัน

โดยเว็บข่าว ให้นศ.ตั้งชื่อ Notebook ว่า

1. news.ipynb
2. quote.ipynb

โค้ดที่นศ. เขียนจะ**ต้อง**มีการเขียนอธิบายที่ชัดเจนโดยใช้ Markdown cells เพื่ออธิบายแต่ละ Code cells ที่อยู่ด้านล่าง (ลำดับถัดไป) ของแต่ละ Markdown cell รวมทั้งใช้ inline (#) or block comments (""" ... """) ประกอบการอธิบายโค้ดและผลลัพธ์ของการวิเคราะห์ข้อมูลของนศ.

โดยทุก ๆ Code cells จะต้องมีการเขียนอธิบายโดย Markdown cells ว่า แต่ละ Code cells นั้นทำอะไรและแปลความหมายผลลัพธ์ว่าอะไร

Part 1: Data Collection (30%)

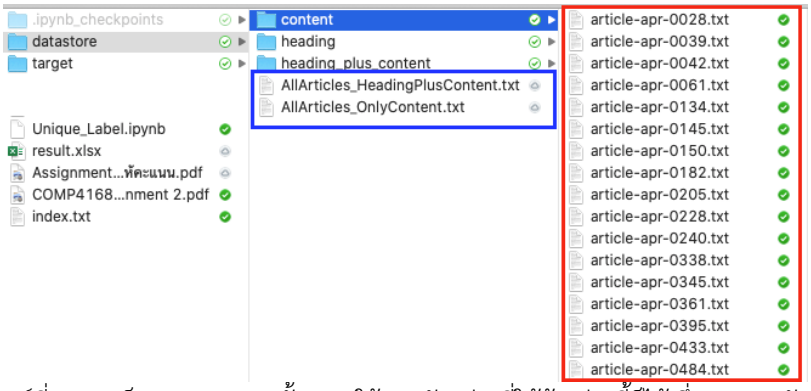
เป้าหมายของส่วนที่ 1 นี้ คือ ให้นักศึกษาฝึกการ scrape ข้อมูลจากหน้าเว็บไซต์ โดยทำการรวบรวมข้อมูล text ที่ถูก labelled (news category or quote tags) แล้ว**ทั้งหมด** ที่ถูกแสดงตามหมวดหมู่ของข่าวหรือการติด tags โดยในส่วนนี้นศ.จะต้องทำรายการงาน ทั้งหมดดังต่อไปนี้

1. ระบุหา URLs และ news category and quote tag labels ทั้งหมด สำหรับข้อมูล text ทั้งหมดที่มีอยู่ในเว็บไซต์ข้างล่างนี้ โดยโค้ดของนศ.จะต้องใช้ root URL เริ่มต้นข้างล่างนี้ในการหาข้อมูล text ที่เหลือทั้งหมด โดยเริ่มจาก URL ตั้งต้นที่ให้

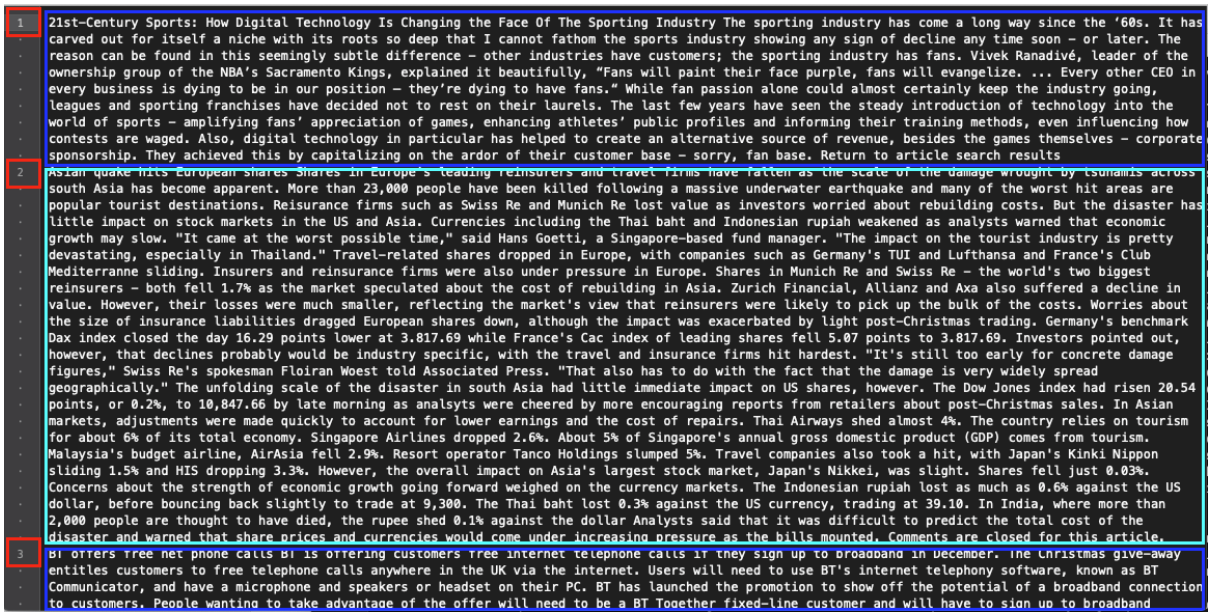
   http://www.it.kmitl.ac.th/~teerapong/news_archive/index.html

   https://quotes.toscrape.com/

2. ให้นศ.ค้นคืนหน้าข้อมูล text ทั้งหมดตาม URLs ที่ได้จากการสกัดจากหน้าเว็บไซต์ที่ได้จากตั้งแต่ root URL ที่ได้ข้างต้นและ crawl ตาม anchor links ที่ปรากฏในหน้าโดย scrape เฉพาะเนื้อหาข้อมูล text ที่จะนศ.จะใช้ในการทดลองเท่านั้น และทำการสกัดส่วนที่เป็น body text ซึ่งมีเนื้อหาของข้อมูล text และทำการ save ส่วนของเนื้อหา text เหล่านั้นเป็น plain text แยกกันของแต่ละเว็บไซต์ โดยแต่ละเว็บไซต์ใช้เพียง 1 ไฟล์ ซึ่งเนื้อข่าวของแต่ละบทความจะถูกบันทึกไว้ในแต่ละบรรทัด (1 บรรทัด : 1 บทความ/quote) โดยให้ทำการ save เนื้อหาข้อมูลไว้ที่ folder ชื่อ ../datastore โดยตั้งชื่อไฟล์ให้สื่อความหมาย และใช้นามสกุล *****.txt เช่น จากรูปตัวอย่างข้างล่าง ผมทำการ crawl ข้อมูล แล้วเก็บไว้ทั้ง 2 แบบ แบบที่เก็บ body text + title และ เฉพาะ body text อย่างเดียว โดยตั้งชื่อไฟล์ว่า “AllArticles_HeadingPlusContent.txt” และ “AllArticles_OnlyContent.txt” และ ตามลำดับ (ในกรอบสีน้ำเงิน)



โดย**ตัวอย่าง**ไฟล์ที่ save เป็น plain text นั้น อาจใช้ตามตัวอย่างที่ให้ข้างล่างนี้ก็ได้ ซึ่ง 1 บรรทัด คือ text ของ 1 บทความ หรือจะใช้ในรูปแบบ csv



3. (optional) heading/title พาดหัวข่าว ของแต่ละบทความ หรือ about author/author detail รายละเอียดของผู้แต่ง สามารถนำมารวมกับส่วนของ body ได้ แต่ถ้านศ.เลือกวิธีนี้ร่วมด้วย ต้องมีกระบวนการในการใช้งานคำที่ปรากฏใน title แตกต่างจาก คำที่ปรากฏอยู่ใน body และมีการทำการทดลองเปรียบเทียบว่า การใช้ title ประกอบในการทำ Text Classification นั้น ช่วยในเพิ่มประสิทธิภาพของการทำ Text Classification โดยใช้คำจาก body เพียงอย่างเดียวหรือไม่ และเพิ่มหรือลดประสิทธิภาพอย่างไร
4. ให้นศ.ทำการบันทึก category/tag labels ของข้อมูล text ทั้งหมด ในไฟล์ที่แยกจาก body เพื่อใช้ในการทำ target variable โดยให้ทำการ save เนื้อหาข้อมูลไว้ที่ folder ชื่อ ../target โดยตั้งชื่อไฟล์ให้สื่อความหมาย และใช้นามสกุล category.txt (1 column) หรือ tags.txt (*n* columns according to the number of total tags)

Part 2: Text Classification (50%)

เป้าหมายของส่วนที่ 2 นี้ คือ ให้นักศึกษาทำการวิเคราะห์จำแนกหมวดหมู่ของข้อความ (text classification) จากข้อมูลจากที่ได้มาใน part 1 โดยในส่วนนี้นศ.จะต้องทำรายการงาน ทั้งหมดดังต่อไปนี้

1. โหลดชุดของไฟล์ที่เราสร้างขึ้นใน part 1 ลงใน Jupyter Notebook ของตัวเอง (ในแต่ละไฟล์ต้องมี class label โดยยึดตาม category/tag label ที่นศ. ได้การ save ไฟล์แยกเอาไว้จากบทความข่าว โดย class label นั้นจะต้องสัมพันธ์กับข่าวหรือ quote ที่ระบุเอาไว้)
2. จากข้อมูล text (raw documents) ที่โหลดมาข้างต้น ให้นศ.สร้าง document-term matrix โดยใช้วิธีที่เหมาะสมในแต่ละขั้นตอน ในการประมวลข้อความเบื้องต้น (text pre-processing) และการถ่วงน้ำหนักของคำ (term weighting) ซึ่งนำไปสู่ประสิทธิภาพของการทำ classification ในลำดับถัดไป พร้อมอธิบายเทคนิคที่นศ.เลือกใช้ในการประมวลผลข้อความเบื้องต้นและการถ่วงน้ำหนักคำ
3. ให้นศ. สร้าง multi-class classification models (ข่าว) และ binary classification problem (quote) อย่างน้อย 2 โมเดล หรือมากกว่า โดยใช้ classifiers ต่างประเภทกัน อย่างน้อย 3 ประเภท หรือมากกว่า **(ยิ่งทำการทดลองครบสมบูรณ์ หลายโมเดล ยิ่งได้คะแนนมาก)** และทำการ Tune โมเดลให้ได้ประสิทธิภาพสูงที่สุด

(จะดูขั้นตอนการ Tune ว่าทำถูกต้องตามหลักวิชาการหรือไม่ หากถูกต้อง และสมบูรณ์ มีวิธีการ Tune ที่ทันสมัย ยิ่งได้คะแนนมาก) และอธิบายเหตุผลของการเลือกประเภทโมเดลมาในการทดลอง พร้อมวิธีการ Tune โมเดลที่คิดว่าจะทำให้ได้ประสิทธิภาพสูงที่สุด

4. ให้นศ. ทำการเปรียบเทียบผลลัพธ์ที่ได้จากแต่ละ models ที่นศ. ได้เลือกไว้ ในข้อ 3 โดยนศ .จะต้องเลือกวิธีกลยุทธ์ หรือการนำเสนอที่เหมาะสมในการประเมินประสิทธิภาพและเปรียบเทียบระหว่างโมเดล นศ. จะต้องรายงานและอภิปรายผลลัพธ์การประเมินที่ได้ลงใน Markdown cells ของ Jupyter Notebook ของตนเอง **(นำผลการการทดลองของแต่ละโมเดลที่ทำการ Tune แล้ว มาเปรียบกันระหว่างโมเดล เช่นรูปแบบตารางหรือ graph ต่าง ๆ ยิ่งสมบูรณ์ ดูทำความเข้าใจง่าย ยิ่งได้คะแนนมาก)**

<u>Code quality and explanation text (20%)</u>

- นศ. ต้องใช้ Markdown cells ในการอธิบายแต่ละขั้นตอนของกระบวนการ โดยนศ. ควรแยกส่วน ของ Part 1 – data collection และ Part 2 – classifier evaluation ให้เด่นชัด แต่ยังอยู่ใน Notebook เดียวกัน ของแต่ละเว็บ
- โค้ดที่นศ. เขียนจะต้องอ่านและเข้าใจได้โดยง่าย ชัดเจน และไม่คลุมเครือ โดยควรมี comment อธิบายที่เพียงพอในการทำให้เข้าใจโค้ดได้โดยง่าย แต่ไม่มากจนเกินไป
- ความซับซ้อนของโค้ดอยู่ในระดับเท่าที่จำเป็น และมีการใช้ Package ต่าง ๆ ที่เหมาะสม โดยเกณฑ์หลัก ๆ จะดูจากวิธีการเซ็ตการทดลองเชิงเปรียบเทียบว่าสมเหตุสมผลหรือไม่ สอดคล้องตามเป้าหมายตามที่นศ. มีเจตนารมณ์ ในการทำหรือไม่ ซึ่งการให้คะแนนจะพิจารณาประสิทธิภาพของ classifier เป็นเรื่องรอง และความเร็วในการประมวลผลเป็นเรื่องรอง (หากการทดลองไม่เวิ่นเว้อจนเกินจุดประสงค์)

<u>Guidelines:</u>

- สำหรับ assignment นี้ อนุญาตให้นศ. ใช้เฉพาะ third-party packages เหล่านี้เท่านั้นในการทำ assignment ได้แก่: **NumPy, Pandas, Scikit-learn, NLTK, SciPy, Requests, BeautifulSoup, Scrapy, Matplotlib, Seaborn** หากใครใช้มากกว่านี้ ต้องขออนุญาตก่อน มิฉะนั้นจะหักคะแนน package ละ 10% และหากมีการอนุญาต ก็จะประชาสัมพันธ์ให้นศ. คนอื่นใช้ได้ด้วย เช่นเดียวกัน
- ให้นศ. ทำการส่ง Assignment ซึ่งคือ
    1. ไฟล์ .ipynb ของ Jupyter Notebook ของนศ. พร้อมข้อมูลที่รวบรวมมา ใน Github repository ของนศ. แต่ละคน โดยในแต่ละ Jupyter Notebook ของนศ. นศ.จะต้องเขียน ชื่อ นามสกุล รหัสนศ. ลงใน Markdown cell แรกของ Notebook
    2. Snapshots ของไฟล์ .ipynb ที่มี output ของทุก cells เป็น
        - ไฟล์ .html โดยการ download as ไฟล์ .ipynb ของนศ. เป็น html
        - ไฟล์ .pdf โดยการ download as ไฟล์ .ipynb ของนศ. เป็น pdf

    <u>ทั้งนี้เพื่อป้องกันการนำไฟล์มารันใหม่แล้วได้ผลลัพธ์ไม่เหมือนเดิม จะได้ตรวจจาก ผลลัพธ์ที่นศ.ทำไว้</u>
- Assignment นี้เป็นงานกลุ่ม โดยแต่ละกลุ่มมีนศ. 1-2 คน หากมีการตรวจสอบพบการคัดลอก (Plagiarism) จะได้ 0 คะแนนในของส่วน Assignment นี้ หากมีข้อสงสัย และหากมีหลักฐานชัดเจนว่ามีการคัดลอกงานจากแหล่งใด ๆ ก็ตาม นศ. จะได้เกรด F ในวิชานี้ และส่งเรื่องต่อให้กับทางคณะฯ และสถาบันฯ ต่อไป

- Hard deadline: วันศุกร์ที่ 27 พฤศจิกายน 2563 เวลา 24.00 น.
    - ส่งช้า 1-5 วัน: ลด 20% จากคะแนนตรวจที่ได้ (ขอเปิด Github ให้ส่งช้า)
    - ส่งช้า 6-10 วัน: ลด 40% จากคะแนนตรวจที่ได้ (ขอเปิด Github ให้ส่งช้า)
    - จะไม่มีการรับตรวจ Assignment หากส่งช้าเกิน 10 วัน โดยปราศจากหลักฐานชี้แจงเหตุผลในการส่งงานช้า ได้แก่ หลักฐานด้านการแพทย์ว่าเข้านอนโรงพยาบาลเพื่อรับการรักษา