**Peem’s ADA Project (Feb 28th 2022)**

**Data description**

We have two datasets:

1. 11,434 videos with content, comments, metadata, etc. without labels on understandability.
2. 600 videos with raw labels on PEMAT guideline on understandability, actionability. A separate label for whether a video contains medical information is included.

Both datasets can be linked by URL (i.e. id) of each Youtube video. Here, I used text to denote human-readable, English phrases with semantic meaning (type = string); string to denote non-language. The dataset consists of the following:

1. captid (string): ID of Youtube caption.
2. captsLastUpdated (text, number)): The last time a video caption is updated in the YYYY-MM-DD: Time format.
3. categoryId (encoding of text): The category each Youtube video belongs to, such as person, blog, science. See Youtube API for more info (<https://gist.github.com/dgp/1b24bf2961521bd75d6c)>.

**\*\*\* NOTE: categoryId differs from relevantTopicId.**

1. channelCommentCount (number): How many comments a video receives.
2. channelDescription (text): Description of each channel (created by channel owner). Found in About section.
3. channelId (text): ID of the video’s channel.
4. channelPublishedat (text): Time the channel was established.
5. channelSubscriberCount (number): How many subscribers a channel has.
6. channelTitle (text): Name of the channel (e.g., All about Diabetes and Related).
7. channelVideoCount (number): How many videos that video’s channel has posted.
8. channelViewCount (number): How many views a channel posting that particular video have received.
9. Comment (text): A list of comments made about each video, stored in a list separated by ,.
10. 'commentCount (text): How many comments have been made on each video.
11. 'contentCaption (Boolean): A Boolean variable saying whether a video has caption or not.
12. 'contentDefinition (string): Consists of two types: hd (high definition) or sd (standard definition). 68% are hd.
13. 'contentDimension (text) : 2d or 3d. Every video except 1 video is in 2d. This feature has no values.
14. 'contentDuration (text, number): Duration of the video, denoted in PT'X'M(Min)'Y'(Sec)
15. 'contentLicensed (text): Whether the content on that channel is licensed. If a video is licensed, it cannot be used for commercial purposes without the permission of the video creator. 67% of the videos are not licensed.
16. description (text): Description of videos seen below the rectangle box (e.g., importance of lipid metabolism).
17. dislikeCount (number): How many dislikes a video receives.
18. embeddable (Boolean): Whether a video can be embedded on (i.e. watched directly from) other websites. Only 1% false; otherwise all true.
19. id (string): id of each video. Also noted as video\_id and URL in some of the datasets.
20. isAutoSynced (Boolean):  Auto Sync refers to automatic syncing of audios and visuals. If a video is auto synced in Youtube, then the video uploader syncs both on Youtube. 67% false. 1% true. Otherwise NaN.
21. keyword (text): Keyword used to search for that video (e.g., diabetes retinopathy)
22. license (text): Is the video licensed by Youtube or Creative Common? 99% by Youtube.
23. likeCount (number): How many likes a video receives.
24. publicStatsViewable (Boolean): Whether users can view public statistics of video. 93% True; 6% false.
25. 'publishedAt' (text, number): Date and time a video is published.
26. rank (number): The rank at which the video appears if searched using keyword in incognito mode.
27. 'relevantTopicIds' (encoding of text): Topic ids created by Youtube based on knowledge graph.
28. 'subtitle' (text): The transcript of what is said in the video (e.g., “In this video, we will talk about”). In this dataset, only 700 videos have subtitles.
29. 'title' (text): Title of the video (e.g., “Importance of insulin”
30. 'topicIds' (encoding of text): All except 3 are NaN
31. 'trackKind' (text): Type of audio track. 1/3 missing. Of the remaining, 87% are ASR and the rest standard.

**(Metadata)**

1. 'viewCount (number): How many views the video receives.
2. 'video\_duration (text, number): Duration of the video
3. 'word\_unique (number)\*\*: Number of unique words in the description of the video.
   1. If the description begins with http://www….., word\_count is automatically set to 0.
4. 'transition\_words (number)\*\*: Number of transition words (think: and, so, but, however, etc.) in the description of the video.
   1. If the description begins with http://www….., word\_count is automatically set to 0.
   2. Many description, though exists, has zero transition words (think: donate here).
5. 'summary\_words (number)\*\*: How many summary words are said.
6. 'active\_verb' (number)\*\*: How many active verbs appear in the description
7. 'sentence\_count' (number) \*\*: How many sentences appear in the description. This is calculated using Python library.
   1. Cannot handle string of words without full stops.
8. 'word\_count' (number) \*\*: How many words appear in the description. This is calculated using Python library.
   1. Cannot handle special symbols.

\*\*\* NOTE: From sentence\_count and word\_count, we can calculate various readability indices. Because these counts are problematic, the following are somewhat unreasonable.

1. 'Kincaid' (number)\*\*: Flesch-Kincaid readability index. Measures the minimum education level required to comprehend the text. The higher Kincaid is, the more difficult a text is to comprehend. See formula using link above.
2. 'FleshReadingEase (number)\*\*:  Flesch reading ease index. Measures how easy it is to read a text. The higher the reading score, the easier a piece of text is to read)
3. 'ARI' (number)\*\*: Automated readability index. The higher ARI is, the more difficult it is to read the text.
4. 'has\_description' (Boolean)\*\*: Whether a video has description.
   1. If the description contains http://, has\_description is automatically set to zero even if it’s followed by an actual text, such as “http://www.handwrittentutorials.com - This video..”
   2. Some videos (around 100) have has\_description = True even though description is empty
5. 'video\_id': Same as id.

**(Derivatives of video metadata)**

1. 'comment\_title\_cosine' (Number):
2. 'comment\_description\_cosine' (Number)
3. 'keyword\_title\_cosine' (Number):
4. 'keyword\_decription\_cosine' (Number)

**(Derivatives of comment)**

1. 'postive\_comment\_count' (Number)
2. 'negative\_comment\_count' (Number)
3. 'comment\_unique\_words' (Number)
4. 'comment\_total\_words' (Number):
5. 'neutral\_comment\_count' (Number):

**Summaries of the descriptive statistics and observations for each variables**

See merge\_and\_cleaned for a description on the PEMAT labels. This is a descriptive statistics on the entire dataset.

**## Useful resource**

1. ARI: https://en.wikipedia.org/wiki/Automated\_readability\_index

2. Flesch reading ease:

3. Kincaid (formally titled: Flesch-Kincaid readability). https://readable.com/readability/flesch-reading-ease-flesch-kincaid-grade-level/

4. Cosine similarity: https://www.machinelearningplus.com/nlp/cosine-similarity/

**## Observations**

1. (Absurdly low readability) The appropriate range of ARI is (-21.43, 14). High ARI = the less readable. Here, 14 denotes a text that should be readable to college students. However, there are 69 videos with ARI > 100. Same issues for Kincaid.

**\*\*Cause:\*\*** ARI is calculated from 0.5(word/sentences). The library counts #sentences based on number of full stops. Therefore, texts with high ARI (very hard to read) are string of words separated by \n: "Yale law school \n Harvard law school ..."

2. (Absurdly high readability) Flesch's appropriateness should be around (100, 0). High Flesch = more readable. High Kincaid = more education required to read the text = less readable.

**\*\*Cause:\*\*** Some description receives extremely high Flesch value because it's a short snippet (e.g., 18740, TNIS).

3. Q: How many readability are out of appropriate range?

A: Around 1,600 (15% of overall dataset), with 900 receiving higher readability than it should be and the rest otherwise.

**\*\*NOTE:\*\*** Flesch reading ease differs from Flesch-Kincaid readability tests (titled Kincaid), which denotes how high your education needs to be to be able to read the text.

4. (Missing) 3,829 videos have no captions, so every field related to caption, such as captionId, trackKind, isLastupdated are also missing.

5. (On cosine similarity) There are 4 cosine similarity values between texts in this dataset (comment/keyword X title/description). 1,664 videos have cosine similarity between comments and something else > 1. Keyword\_title and Keyword\_description do not experience this problem.

**\*\*Cause:\*\***

* Comments are stored as list, but are represented in the string form. E.g., "[Wow that's great, LOL]". Python interprets the left bracket as special symbols.
* 2) Special symbols such as !\*.

6. (On commentCount) I don't understand the relationship between positive/negative/neutral comment count. The sentiment analysis should assign positive/negative/neutral label to each of the comment, so the sum of these three should be equal to commentCount. However, there are 500 rows where commentCount drastically exceeds the sum (e.g., 319,810 comments vs 300 as sum).

\*\***Cause(?):\*\*** Special symbols?

7. (On channels). Three variables related to channels are channelSubscribers, channelViewCount, channelCommentCount. I have checked channels with very high subscribers/views/comments. Most of them are legitimate, such as Khan Academy, CBS, CNN. However, the distribution is \*\*extremely\*\* right-skewed (mean >>>> median). See histogram in R. Around 4,000 channels have no channel description.

8. (Rank) Denotes the rank the video appears.

9. (Comment,views, likes, duration) Like-dislike-view and duration are all right-skewed. The former is related to one another, so \*\*I intend to perform some PCAs\*\* on these variables.

10. (Weird discovery)

* 10.1. The video with highest commentCount is Skrillex — First of the Year Equinox. It's an EDM song that has nothing to do with diabetes. <https://www.youtube.com/watch?v=2cXDgFwE13g>
* 10.2 The channel with highest subscriber count is Canal Kondzilla, a Spanish rapper. <https://www.youtube.com/results?search_query=Canal+KondZilla>
* 10.3 There's a Bollywood movie about reincarnated Hindu goddess (see channelSubscriberCount > 20m).
* 10.4. From checking videos with comment\_unique\_word > 4,000, I found out several duplicates of the ID already. They all appear in top results for different keywords.

**\*\*Cause\*\*:** Keywords such as 'nesina', 'starlix' refer to diabetes drugs. However, if you search these keywords on Youtube, you sometimes see Youtube channels/titles with these names.

**## Questions to Nynke and Larry**

**1.** Every derivative of comments is problematic because it contains special symbols, https://, etc. I'm thinking about spending a huge amount of time cleaning the comments and recomputing their derivatives, such as negative/positive comment counts. Is this a useful step to take?

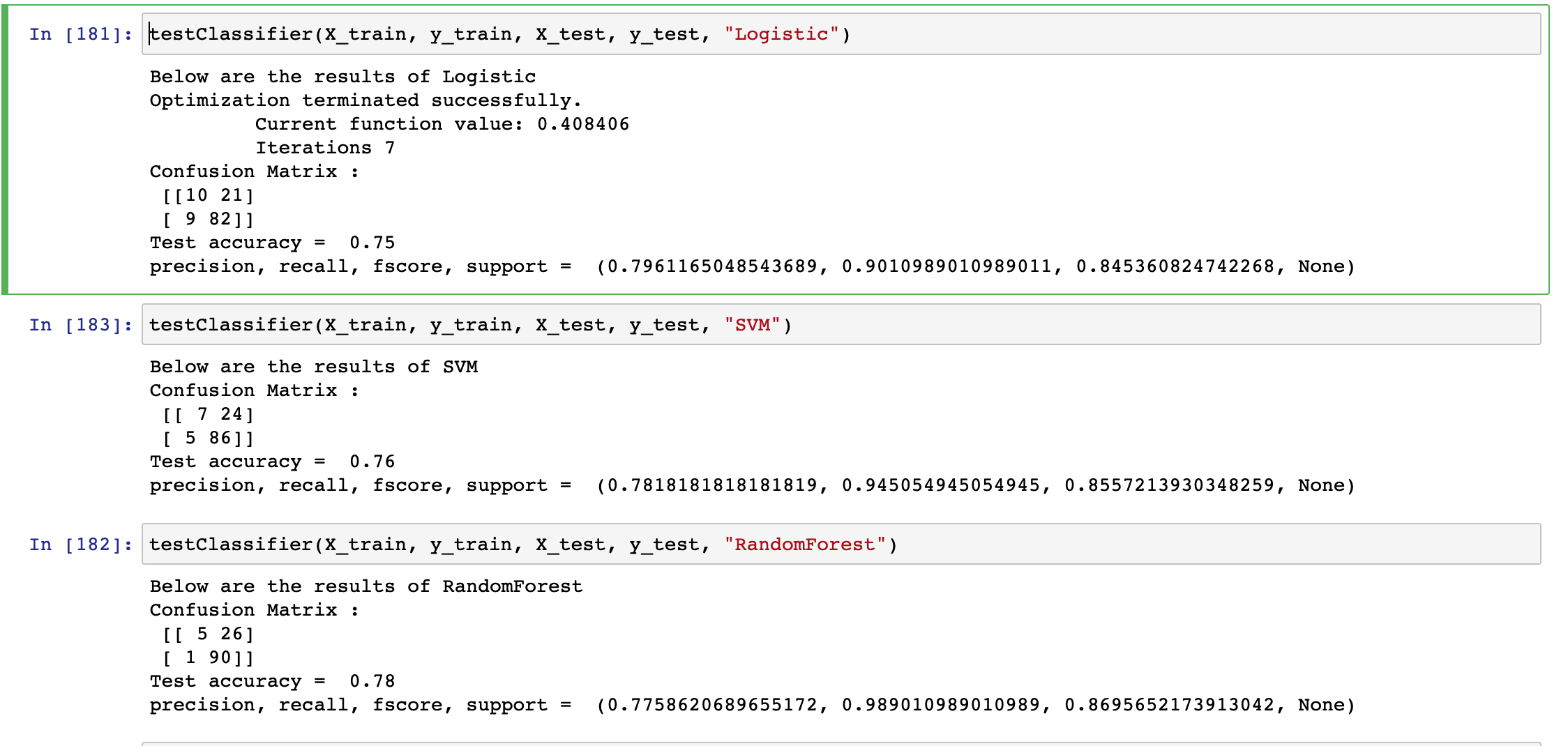
2. This dataset contains quite a lot of irrelevant videos (see the distribution of relevantTopicIds and weird discovery). My proposal is to remove a lot of videos based on Youtube's topicId labels. ARound 900 videos will be removed. What do you think of this approach?

3. Many diabetes videos appears several times under different tags. These tags are definitely useful. However, my concern is that the co-training would label same video with different labels, especially on videos it doesn't predict with high confidence. How much will this affect co-training performance, if any? Any preventive measures necessary?

4. I have compared three models: logistic regression, SVM, randomForest (see below). Overall, there’s negligible difference in accuracy where logistic regression performs slightly worse than SVM, randomForest, so it’s very unclear which model to choose based on these metrics.

What’s more concerning is that I could not replicate the results on Page 23 of Rema’s paper: <https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3711751>

My model includes fewer variables (i.e. no transcription confidence, no transcript medical term, no Boolean on tags/titles/description), so my model’s performance should be worse than what appears on the paper. The opposite happens.



I have set the seed and used approximately the same numbers in the evaluation set. Therefore, the difference in performance comes from something else (what?).

5. Co-training relies on the assumption that two sets of features are conditionally independent but can be used to predict equally well. There are a few videos where the description is exactly the transcript (from EDA), which definitely violates the assumption. Some transcripts contain absurd values (e.g., 19454) that shouldn’t be able to predict any labels. Your thoughts on how we might extend this?

6. Does co-training work with duplicates??

**## Extra stuff**

I spent some time thinking about the network-based approach. Many papers use channel-to-channel subscription and/or the number of referrals, both of which we don't have. However, there are several ways we may connect videos together, such as if they appear on the Top 50 using same keywords, if they have same topicId. TopicIds are shown below:

{'Society (parent topic)': 5812,

'Knowledge': 1401,

'Lifestyle (parent topic)': 5638,

'Health': 2035,

'Food': 580,

'Fitness': 193,

'Music (parent topic)': 1162,

'TV shows': 216,

'Movies': 166,

'Entertainment (parent topic)': 643,

'Technology': 158,

'Gaming (parent topic)': 94,

'Electronic music': 41,

'Hobby': 48,

'Vehicles': 31,

'Sports (parent topic)': 25,

'Rock music': 53,

'Pop music': 35,

'Hip hop music': 54,

'Military': 5,

'Politics': 12,

'Christian music': 4,

'Religion': 8,

'Performing arts': 65,

'Reggae': 7,

'Soul music': 3,

'Action-adventure game': 1,

'Role-playing video game': 17,

'Action game': 38,

'Pets': 49,

'Rhythm and blues': 1,

'Golf': 2,

'Cricket': 1,

'Fashion': 4,

'Simulation video game': 2,

'Music of Asia': 7,

'Motorsport': 4,

'Basketball': 7,

'Independent music': 16,

'Racing video game': 2,

'Music of Latin America': 23,

'Sports game': 3,

'Football': 1,

'Strategy video game': 1,

'Business': 3,

'Humor': 2,

'Boxing': 2,

'Classical music': 1,

'Jazz': 1,

'Baseball': 2}

Without proper edge weighting, the output from network-exposure-model is the majority.